

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu prawidłową odpowiedzią jest n.
4. Tu też prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. O, a tutaj t.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2~ EE3)* EE4`, gdzie `~` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu też prawidłową odpowiedzią jest n.
3. Tu prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tutaj znowu n.
3. O, a tutaj t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tu prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. A tu prawidłową odpowiedzią jest t.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną kod. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2~ EE3)* EE4`, gdzie `~` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. A tu prawidłową odpowiedzią jest t.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. O, a tutaj t.
3. A tu prawidłową odpowiedzią jest t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tutaj znowu n.
3. A tu prawidłową odpowiedzią jest t.
4. Tu też prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. O, a tutaj t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tu prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n .
2. Tu też prawidłową odpowiedzią jest n .
3. A tu prawidłową odpowiedzią jest t .
4. Tu prawidłową odpowiedzią jest n .
5. O, a tutaj t .

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. O, a tutaj t.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tutaj znowu n.
3. A tu prawidłową odpowiedzią jest t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tutaj znowu n.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. O, a tutaj t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tutaj znowu n.
3. A tu prawidłową odpowiedzią jest t.
4. O, a tutaj t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tu prawidłową odpowiedzią jest n.
4. Tu też prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. A tu prawidłową odpowiedzią jest t.
3. O, a tutaj t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tutaj znowu n.
4. Tu też prawidłową odpowiedzią jest n.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu prawidłową odpowiedzią jest n.
3. O, a tutaj t.
4. Tu też prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tu prawidłową odpowiedzią jest n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2~ EE3)* EE4`, gdzie `~` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. O, a tutaj t.
3. A tu prawidłową odpowiedzią jest t.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tu prawidłową odpowiedzią jest n.
4. Tu też prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. O, a tutaj t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tutaj znowu n.
2. A tu prawidłową odpowiedzią jest t.
3. O, a tutaj t.
4. Tu prawidłową odpowiedzią jest n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. A tu prawidłową odpowiedzią jest t.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu też prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. Tu prawidłową odpowiedzią jest n.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tu też prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. O, a tutaj t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu też prawidłową odpowiedzią jest n.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. O, a tutaj t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. A tu prawidłową odpowiedzią jest t.
3. Tutaj znowu n.
4. O, a tutaj t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. A tu prawidłową odpowiedzią jest t.
4. Tutaj znowu n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie `^` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. Tu prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. A tu prawidłową odpowiedzią jest t.
2. O, a tutaj t.
3. Tu też prawidłową odpowiedzią jest n.
4. Tu prawidłową odpowiedzią jest n.
5. Tutaj znowu n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2 \wedge EE3)* EE4`, gdzie \wedge oznacza permutację linijek, $*$ oznacza powtórzenie, a k definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, którzy należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. A tu prawidłową odpowiedzią jest t.
3. Tu prawidłową odpowiedzią jest n.
4. Tutaj znowu n.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: L^AT_EX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EEn` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2~ EE3)* EE4`, gdzie `~` oznacza permutację linijek, `*` oznacza powtórzenie, a `k` definiuje L^AT_EX-ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. Tu prawidłową odpowiedzią jest n.
2. Tutaj znowu n.
3. Tu też prawidłową odpowiedzią jest n.
4. O, a tutaj t.
5. A tu prawidłową odpowiedzią jest t.

Powodzenia!

Program miesza pytania w egzaminie, w sensie każdy student dostaje trochę inną kartkę, różniącą się (jedynie) kolejnością pytań. Każde pytanie ma odpowiedź jednoliterową (np. t lub n). Każda kopia egzaminu oznaczana jest kodem składającym się z rzeczownika i przymiotnika. Sprawdzanie jest bardzo proste: sprawdzający tworzy plik tekstowy, którego każda linijka zawiera dwuliterowy kod (pierwsza litera rzeczownika i przymiotnika), ciąg odpowiedzi studenta, jego nazwisko i imię. Program znajduje, która z kopii egzaminu była oznaczona danym kodem, i wypisuje, ile odpowiedzi poprawnych i ile niepoprawnych udzielił każdy ze studentów.

Wymagania: \LaTeX i C++. Najwygodniej używa się pod środowiskiem unixowym.

Przygotowanie egzaminu: Obejrzyj plik `exmix.tex`. Składa się on z kilku sekcji, oznaczonych komentarzem `EE n` (dla n od 0 do 6). Wytworzony plik będzie miał postać: `EE0 (k EE1 EE2^ EE3)* EE4`, gdzie [^] oznacza permutację linijek, * oznacza powtórzenie, a `k` definiuje \LaTeX -ową zmienną `kod`. Sekcje `EE5` i `EE6` definiują rzeczowniki i przymiotniki. Przeedytuj plik według własnych potrzeb, uruchom polecenie `make`. Zostanie stworzony plik `egzamin.pdf`, który należy obejrzeć i wydrukować.

Ocenianie egzaminu: wpisz odpowiedzi studentów w pliku `odp.txt`. Uruchom polecenie `make` jeszcze raz. W pliku `wyniki.txt` będzie wypisana liczba prawidłowych i nieprawidłowych odpowiedzi dla każdego studenta, a w pliku `statystyki.tex` statystyki trudności dla poszczególnych pytań.

Przykładowe pytania:

1. O, a tutaj t.
2. Tu prawidłową odpowiedzią jest n.
3. Tutaj znowu n.
4. A tu prawidłową odpowiedzią jest t.
5. Tu też prawidłową odpowiedzią jest n.

Powodzenia!