

Praca zaliczeniowa

Praca zaliczeniowa powinna składać się z:

1. Programu napisanego w C++ realizującego założone zadanie.
2. Dokumentacji do programu. Dokumentacja powinna zawierać:
 - opis zadania realizowanego przez program
 - opis funkcji programu
 - instrukcję użytkownika programu
 - krótki opis kodu programu.
3. Opisu eksperymentów (opcjonalnie). Jeśli tematem programu jest przeprowadzenie pewnych symulacji, należy opisać je i wyciągnąć z nich wnioski.

Praca powinna być wykonana przez zespoły, najlepiej 2-3 osobowe. Przed rozpoczęciem pracy należy zgłosić jej temat do zaakceptowania. Gotowe prace należy wysłać na adres ziemians@mimuw.edu.pl do 20 czerwca.

Propozycje tematów na program zaliczeniowy

Kalkulator

Napisać program, który umożliwia wyliczanie wartości wyrażeń arytmetycznych wpisywanych do terminala. Należy zadbać o właściwą kolejność działań. Możliwe rozszerzenia: definiowanie własnych zmiennych i funkcji.

Automaty komórkowe

Stworzyć symulator gry z życia https://pl.wikipedia.org/wiki/Gra_w_życie. Plansza podzielona jest na kwadratowe pola. W niektórych polach znajdują się komórki. Jeśli komórka ma mniej niż dwóch lub więcej niż trzech sąsiadów, znika; jeśli wolne pole ma dokładnie trzech sąsiadów, pojawia się na nim nowa komórka. Program powinien umożliwiać przeprowadzenie symulacji dla różnych początkowych układów.

Możliwe warianty: inne gry komórkowe, np. mrówki Langtona.

Układanka

Dany jest pewien wzorec składający się z przylegających do siebie kwadratów o jednakowych rozmiarach, oraz pewien zestaw klocków (również składających się z kwadratów). Należy stworzyć program, który znajduje pokrycia wzorca (jedno lub wszystkie) przy pomocy nienachodzących na siebie klocków. Możliwe warianty:

rozpatrywać siatki trójkątne lub sześciokątne. Więcej szczegółów:
np. <http://en.wikipedia.org/wiki/Pentomino>.

Dylemat więźnia

Pewna ilość graczy (n) w każdej rundzie gra w dylemat więźnia, każdy z każdym pewną ilość rund (l). Po każdej rundzie kilku graczy (k), którzy uzyskali najgorsze wyniki umiera, a ich miejsce zajmuje potomstwo k najlepszych graczy. Każdy z graczy stosuje ustaloną strategię - dokonuje wyboru z prawdopodobieństwem określonym tylko i wyłącznie na podstawie ostatnich r ruchów przeciwnika. Potomek zwycięskiego gracza otrzymuje nieznacznie zaburzoną strategię rodzica. Należy napisać program, który wykonuje symulacje i umożliwia dokonanie ich analizy (tj. stwierdzenie, jakie strategie odniosły sukces).

Można też rozważać inne gry, np. papier-kamień-nożyce.

Korelacje

Należy stworzyć program, który wyszukuje korelacje pomiędzy zmianami kursów akcji w poszczególnych dniach. Przetestować program i wyciągnąć wnioski przy użyciu rzeczywistych danych. Możliwe warianty: przetestować w podobny sposób inne dane, np. meteorologiczne.

Kompresja tekstów

Zaimplementować program wykonujący kompresję i dekompresję tekstów metodą Huffmana (zob. http://en.wikipedia.org/wiki/Huffman_coding). Opracować optymalne kody do kodowania tekstów w różnych językach. Sprawdzić, czy zapisywanie niektórych często występujących słów jako pojedynczych symboli zwiększa efektywność kodowania.