

1. Napisać funkcję

```
int tylkoDodatnie(int* t, int rozmiar, int* d)
```

która przepisuje (tylko) dodatnie elementy tablicy `t` o rozmiarze `rozmiar` do tablicy `d` (zakładamy, że `d` jest dostatecznie duża). Kolejność elementów powinna zostać zachowana, a wynik powinien być liczbą elementów przepisanych.

2. Napisać funkcję

```
bool znajdzRowne(int* t, int rozmiar)
```

która zwraca `true` wtedy i tylko wtedy, gdy pewien element w tablicy `t` się powtarza.

3. Napisać funkcję

```
void wspak(int* t, int rozmiar)
```

która odwraca tablicę `t` o rozmiarze `rozmiar` (tzn. pierwszy element ma znaleźć się na końcu, drugi na przedostatnim miejscu, itd.)

4. Napisać funkcję

```
int sumaDzielnikow(int n)
```

która zwraca sumę wszystkich dzielników liczby `n`.

5. Napisać funkcję

```
void usunSpacje(char* s, char* t)
```

która umieszcza w tablicy `t` napis `s` z usuniętymi spacjami. Zakładamy, że w `t` jest dostatecznie dużo miejsca.

6. Napisać funkcję

```
int pierwiastek(int n)
```

która zwraca pierwiastek z liczby `n` zaokrąglony w dół do liczby całkowitej.

7. Napisać funkcję

```
int sumaRoznych(int* t, int rozmiar)
```

która zwraca sumę **różnych** elementów tablicy `t` o rozmiarze `rozmiar`.

8. Napisać funkcję

```
int fibo(int n)
```

która zwraca n -tą liczbę Fibonacciego (tj. $f(0)=f(1)=1$, $f(n)=f(n-1)+f(n-2)$ dla $n>1$).

9. Napisać funkcję

```
int drugi(int* t, int rozmiar)
```

która zwraca drugi największy element w tablicy `t` o rozmiarze `rozmiar`. W przypadku, gdy są dwa równe największe elementy należy zwrócić jeden z nich. Można założyć, że `rozmiar > 1`.

10. Napisać funkcję

```
void przesun(int *t, int rozmiar)
```

która przesuwa w prawo elementy tablicy `t` o rozmiarze `rozmiar`. Ostatni element powinien znaleźć się na początku.

11. Podać wynik działania programu

```
int main() {
    int l=0;
    for(int x=5; x; x--) {
        for(int y=x; y; y--)
            for(int z=y; z; z--)
                l++;
        cout << l << endl;
    }
}
```

12. Podać wynik działania programu. Wyjaśnić, co zwraca funkcja `f` dla dowolnego parametru.

```
int f(char* s) {
    if (*s)
        return 2*f(++s);
    return 1;
}

int main() {
    cout << f("Ala") << endl << f("Eugeniusz") << endl;
}
```

13. Podać wynik działania programu

```
int main() {
    int x=8;
    for(int i=0; i<x; ++i) {
        for(int j=0; j<x; j++) {
            if((i+j)%2)
                cout << j;
            else
                cout << 'X';
        }
        cout << endl;
    }
}
```

14. Podać wynik działania programu. Wyjaśnić, co zwraca funkcja f dla dowolnego parametru.

```
void f(int n) {
    if(n) {
        int a;
        if (a=n%10) {
            cout << a;
            f(n-1);
        }
        else
            f(n/10);
    }
}

int main() {
    f(35); cout << endl; f(825); cout << endl; f(1000);
}
```

15. Podać wynik działania programu

```
int f(int a, int b) {
    cout << a << ',' << b << endl;
    if (a>0)
        return 1 + f(a+b, b-1);
    return 0;
}

int main() {
    cout << f(21, 0) << endl << f(1, 5) << endl << f(30, -10) <<
endl;
}
```

Czy funkcja f zawsze (tj. dla każdego parametru) zakończy działanie? Odpowiedź uzasadnić.

16. Podać wynik działania programu

```
int main() {
    int t[10];
    for(int i=0; i<10; i++)
        t[i]=i+1;
    for(int i=9; i; --i)
        for(int j=i-1; j+1; --j)
            t[i] += t[j];
    for(int i=0; i<10; i++)
        cout << t[i] << endl;
}
```

17. Podać wynik działania programu

```
int main() {
    char* s = "Alicja w krainie czarow";
    for(char* p=s; *p; p++) {
        cout << *p;
    }
}
```

```

        if (*p==' ')
            cout << p[1];
    }
}

```

18. Podać wynik działania programu

```

int f(int n, int d) {
    if (n==1)
        return 1;
    if(n%d)
        return f(n,d+1);
    cout << d << ',';
    return d*f(n/d, d);
}
int main() {
    int a = f(420, 2);
    cout << endl << a;
    a = f(77, 2);
    cout << endl << a;
}

```

19. Podać wynik działania programu

```

int main() {
    int x=7;
    while(x--) {
        for(int y=0; y<=x; ++y)
            cout << y;
        cout << endl;
    }
}

```

20. Podać wynik działania programu. Wyjaśnić, co zwraca funkcja f dla dowolnego parametru.

```

int f(int n) {
    if (n)
        return n%2 + f(n/2);
    return 0;
}
int main() {
    cout << f(10) << endl << f(31) << endl << f(256) << endl;
}

```