

Spis treści

1.	Pętle	2
2.	Funkcje	5
3.	Wskaźniki	9
4.	Struktury, klasy, unie	

Pętle

Zadania dotyczące programów należy wykonać przy użyciu pętli `for`, `while` i `do ... while`.

1. Uzupełnij poniższe zapisy tak by pętle wykonały się `n` razy:

- a) `n = 1,`
`for (int i = ... ; i < 10; i++) cout << i;`
- b) `n = 5,`
`for (int i = 1 ; i < ... ; i++) cout << i;`
- c) `n = 10,`
`for (int i = 1 ; i ... ; i--) cout << i;`
- d) `n = 20,`
`for (int i = 10 ; i ... ; i ...) cout << i;`
- e) `n = 7;`
`for (int i = 0 ; i ... ; i--) cout << i;`

2. Zapisz w postaci blokowej algorytm realizujący pętle:

- a) `for`,
- b) `while`,
- c) `do...while`.

3. Napisz program, który zapyta użytkownika o imię i wyświetli je 10 razy.

4. Napisz program, który wyświetli liczby naturalne z przedziału `<1;100>` oddzielając je spacją.

5. Zrealizuj program obliczający sumę `n`-wczytanych liczb.

6. Zrealizuj program pobierający od użytkownika `n`-liczb i podający ile wśród nich było większych od zera.

7. Napisz program wyświetlający na ekranie kolejne liczby całkowite typu `int`, które są podzielne bez reszty przez `n` (gdzie `n` jest zadawane z klawiatury, a zakres wyświetlanych liczb to `<1;1000>`).

8. Napisz program wczytującą kolejne liczby z klawiatury i kończącą się gdy:

- a) suma tych liczb przekroczy 100,
- b) ilość podanych liczb ujemnych przekroczy 5,
- c) dwie kolejne podane liczby będą miały identyczną wartość.

9. Napisz program wypisujący na ekranie liczby od `X` do 0 gdzie `X` jest liczbą całkowitą i podawany jest przez użytkownika.

10. Podaj możliwie najprostszy sposób spowodowania, że program zawierający pętlę `while( )` zacznie się zachowywać się

jak program zawierający pętlę do ... while() (tzn. pętla zawsze się wykona przynajmniej jeden raz). Czy istnieje analogiczna możliwość zamiany rolami pętli do ... while() za pomocą pętli while()? Podaj przykład.

11. Napisz program, który wczyta dane z klawiatury do 10-cio elementowej tablicy liczb double, a następnie sprawdzi:

- czy te liczby są uporządkowane rosnąco lub malejąco,
- czy elementy tablicy mają symetryczną zawartość (tzn. pierwszy element równy ostatniemu, drugi równy przedostatniemu, itd.),
- czy w tablicy występuje chociaż jedno powtórzenie wartości.

12. Napisz program, który będzie wczytywał i analizował tekst (ciąg liter) wprowadzany z klawiatury aż do momentu wpisania znaku "0" (zero). Po zakończeniu wczytywania program powinien wyświetlić na ekranie histogram występowania poszczególnych liter alfabetu np.

```
A 17 #####
B  2  ##
C  1  #
D  0
E  5  #####
...
Z  3  ###
```

Do gromadzenia informacji o ilości podanych liter wykorzystaj tablicę 26 „liczników”.

13. Napisz program wyświetlający liczby z zakresu <65;90>. Obok każdej z nich wyświetl odpowiadający jej znak z tablicy ASCII.

```
65 A
66 B
...
```

14. Napisz program, który pobierze od użytkownika liczbę naturalną z przedziału <1;1000>. Następnie wyświetli 30 kolejnych liczb parzystych większych od wartości podanej przez użytkownika.

15. Sformułuj program wyliczający średnią arytmetyczną z ciągu liczb podawanych przez użytkownika, zakończonego liczbą 0. Liczby 0 nie bierz pod uwagę w średniej.

16. Napisz program, który poprosi użytkownika o podanie dwóch liczb. Program ma za zadanie zsumować wszystkie liczby całkowite z przedziału podanego przez użytkownika. Załóżmy, iż pierwsza liczba jest mniejsza od drugiej, czyli np. 2 i 4, suma to 9 (czyli 2 + 3 + 4).

17. Napisz program, który dla podanego n obliczy następujące wartości

a) $s=1^2+2^2+3^2+\dots+n^2$

b) $s=1^2-2^2+3^2-\dots+(-1)^n n^2$

c) $s=n!=1*2*3*\dots*n$

d) $s=1/2+2/3+3/4+5/6+\dots+n/(n+1)$

e) $s=(1^2+1)/2^2+(2^2+1)/3^2+\dots+(n^2+1)/(n+1)^2$

18. Napisz program, który stworzy następujący rysunek (poprzez wyświetlanie znaku x i spacji) - wysokość rysunku powinna być wczytywana z klawiatury.

a)	b)	c)
xxxx xxxx xxxx	x xx xxx	xxx xx x
d)	e)	f)
x xx xxx	xxxxx xxx x	x xx xxx xx x

19. Napisz program, który wyświetli tabliczkę mnożenia (ilość elementów w wierszu/kolumnie podaje użytkownik).

20. Napisz program, który utworzy tablicę 5x5 i wypełnij ją jedynekami. Utworzoną tablicę wyświetl na ekranie.

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

21. Napisz program tworzący tabelę, w której na głównej przekątnej znajdują się kolejne liczby naturalne a poza przekątną same jedyńki. Rozmiar tablicy pobierz od użytkownika. Utworzoną tabelę wyświetl na ekranie.

1	1	1	1	1
1	2	1	1	1
1	1	3	1	1
1	1	1	4	1
1	1	1	1	5

22. Napisz program, który, zakładając, że Ziemia to idealna kula o promieniu 6371 km, wypisze odległość do linii horyzontu od punktu o wysokości 1 km, 2 km, . . . , 10 km nad Ziemią.

Funkcje

Wszystkie stworzone funkcje należy wywołać z głównej funkcji programu.

1. Stwórz funkcję, która wypisze na ekran słowo "witaj".
2. Stwórz funkcję, która jako parametr przyjmuje imię użytkownika i wyświetla je na ekranie.
3. Stwórz funkcję, która pobierze od użytkownika imię i wyświetli je 5 razy;
4. Stwórz funkcję przyjmującą dwa całkowite parametry i wyświetlającą wszystkie liczby parzyste mieszczące się między przekazanymi do funkcji wartościami.
5. Stwórz funkcję, która oblicza iloraz dwóch liczb całkowitych. Liczby są przekazywane do funkcji jako jej argumenty.
6. Stwórz funkcję, która przyjmuje dwa parametry rzeczywiste. Jej zadaniem ma być zwrócenie większej z dwóch otrzymanych liczb.
7. Stwórz funkcję, która przyjmie dwa parametry naturalne. Zadaniem funkcji jest zamiana liczby dziesiętnej na binarną na odpowiedniej ilości bitów. Pierwszy parametr to długość reprezentacji binarnej a drugi to liczba do zamiany. Funkcja ma zwracać ciąg znaków zawierający odpowiednio przygotowany ciąg zer i jedynek.
8. Stwórz funkcję, która dla zadanej tablicy dwuwymiarowej sprawdzi w niej istnienie wartości. Tablica i szukana wartość mają być parametrami funkcji. Funkcja zwraca -1 gdy nie ma szukanej wartości w tablicy, 1 gdy szukana wartość jest w tablicy.
9. Stwórz funkcję, która odpowie 1 gdy przekazany jej rok (np. 1999) jest przestępny lub 0 gdy nie jest przestępny.
10. Stwórz funkcję, która przyjmie dwie daty w postaci ciągu znaków np. 13.11.2011 i zwróci w postaci liczby całkowitej informację o liczbie dni różniacej te dwie daty.
11. Stwórz funkcję, która jako parametr przyjmie trzy liczby

całkowite. Zadaniem funkcji ma być sprawdzenie czy podane liczby są liczbami pitagorejskimi (funkcja ma zwrócić 1 gdy odpowiedź jest twierdząca, 0 w przeciwnym wypadku).

12. Stwórz funkcję zmieniającą położenie konika szachowego. Jako argumenty funkcja otrzymuje 2 współrzędne aktualnego położenia konika. Jako wynik zmienione zostają obydwie współrzędne, ale w taki sposób, by ruch był dozwolony.

13. Stwórz funkcję, która jako parametr przyjmie tekst. Zadaniem funkcji ma być sprawdzenie, która litera w tekście pojawia się najczęściej. Funkcja ma zwrócić literę (tę, która występuje najczęściej).

14. Stwórz funkcje, które jako parametr przyjmą (każdy podpunkt to osobne zadanie/funkcja)

- a. liczbę całkowitą,
- b. ciąg znaków,
- c. liczbę rzeczywistą.

Zadaniem funkcji ma być sprawdzenie czy podana liczba jest palindromem. Funkcje mają zwrócić 1 gdy w parametrze otrzymała palindrom, 0 w przeciwnym wypadku.

15. Stwórz funkcję, która jako parametr przyjmuje liczbę rzeczywistą. Zadaniem funkcji ma być zwrócenie wartości bezwzględnej podanej liczby.

16. Stwórz funkcje, które wyznaczają wartości

- a. $n!$,
- b. 2^n ,
- c. a^n ,

dla a i $n \in \mathbb{N}$.

Wyznaczone wartości mają być zwrócone przez funkcję. Funkcje przygotuj w formie iteracyjnej oraz rekurencyjnej.

17. Stwórz funkcję, która przyjmuje jako parametr liczbę rzeczywistą. Zadaniem funkcji ma być zwrócenie pierwiastka kwadratowego z podanej w parametrze wartości z dokładnością $\epsilon=0,001$. (bez użycia wbudowanych funkcji matematycznych)

18. Stwórz funkcję, która zwróci wartość n -tego elementu w ciągu Fibonacciego (iteracyjnie i rekurencyjnie).

19. Stwórz funkcję, która jako parametr przyjmie liczbę naturalną n ($0 < n < 10^6$). Zadaniem funkcji jest zwrócenie 1 gdy

podana liczba jest liczbą pierwszą a 0 w przeciwnym wypadku.

20. Stwórz funkcję, która jako parametr przyjmie liczbę naturalną n ($0 < n < 10^6$). Zadaniem funkcji jest rozłożenie liczby na czynniki pierwsze i zapisanie wyniku na standardowym wyjściu w postaci $n = p_1 * p_2 * p_3 * \dots * p_i$ gdzie n to podana liczba a p_1 do p_i to liczby pierwsze.

21. Stwórz funkcję generującą liczby pseudolosowe. Liczba inicjująca niech będzie wartość z przedziału $(0,1)$. Kolejne wartości niech będą wyznaczane ze wzoru $x_n = 1 - x_{n-1}^2$ gdzie x_{n-1} to poprzednio wygenerowana liczba.

22. Napisz funkcję, która dostaje jako argumenty pięć liczb typu unsigned int i zwraca jako wartość sumę podanych liczb. Funkcję napisz w taki sposób, żeby liczyła sumę także dwóch, trzech i czterech argumentów.

23. Napisz funkcję, która przyjmie dwie liczby naturalne. Zadaniem funkcji jest zwrócenie:

- a. największego wspólnego dzielnika,
- b. najmniejszej wspólnej wielokrotności.

24. Napisz funkcję, która przyjmie jako parametr literę i zwróci odpowiadającą tej literze liczbę z tablicy kodów ASCII.

25. Napisz funkcję, która przyjmie trzy parametry - liczby naturalne a , b i c . Funkcja ma zadanie zwrócić wynik działania $(a + b) \bmod c$.

26. Napisz funkcję, która jako parametr przyjmie znak. Jeżeli ten znak jest cyfrą funkcja ma zwrócić jej wartość np. '5' -> 5, w przeciwnym przypadku funkcja ma zwrócić -1.

27. Napisz funkcję, która jako parametr przyjmie ciąg znaków s . Funkcja ma zadanie odwrócić ciąg znaków (pierwsza litera staje się ostatnia, ostatnia pierwszą itd.) i zwrócić wynik swojej pracy.

28. Napisz funkcję, która przyjmie jako parametr liczbę naturalną n . Zadaniem funkcji jest wyznaczenie i przekazanie n -tego wyrazu ciągu określonego wzorem:

- a. $a_n = \{ \begin{matrix} 3 & \text{dla } n = 1, \\ a_{n-1} + 10 & \text{dla } n > 1, \end{matrix}$
- b. $a_n = \{ \begin{matrix} 0,2 & \text{dla } n = 1, \\ a_{n-1} * n & \text{dla } n > 1, \end{matrix}$
- c. $a_n = \{ \begin{matrix} 2/3 & \text{dla } n = 1, \\ a_{n-1} * 1/2^n & \text{dla } n > 1, \end{matrix}$

gdzie $n \in \mathbb{N}$.

29. Napisz funkcje, które przyjmą jako parametry dwie liczby binarne (długość reprezentacji do 128 bitów) i wykona na nich następujące działania (każdy podpunkt oznacza osobną funkcję):

- a. suma,
- b. różnica,
- c. mnożenie,
- d. dzielenie całkowite,
- e. dzielenie modulo

i zwróci wynik w postaci binarnej.

30. Napisz funkcję, która jako parametr przyjmie liczbę całkowitą. Zadaniem funkcji ma być zwrócenie reprezentacji u2 (uzupełnienie do dwóch) tej liczby w 16. bitowej reprezentacji. Określ największy przedział do jakiego może należeć przekazany do funkcji parametr.

31. Napisz funkcję, która jako parametr przyjmie liczbę rzeczywistą i zwróci jej reprezentację binarną w postaci $a_{n-1}...a_2a_1a_0,a_{-1}a_{-2}a_{-3}...a_{-m}$ gdzie n i m są liczbami naturalnymi oznaczającymi pozycję bitu. (przyjmij 16. bitową reprezentację gdzie $n=12$ i $m=4$)

32. Napisz funkcję, która przyjmie dwa parametry:

1. tablica,
2. rozmiar tablicy.

Funkcja ma wypełnić przekazaną tablicę losowymi elementami.

33. Przeanalizuj kod programu:

```
#include <iostream>

using namespace std;

int tab[5000];

int aa(int b);

int memorize(int (*a)(int b),int n)
{
    if(tab[n]==0)
        tab[n] = aa(n);
    return tab[n];
}
```

```
int aa(int b)
{
    if(b == 1 || b == 0) return 1;
    return memorize(aa, b-1)+memorize(aa, b-2);
}

int main()
{
    cout << memorize(aa,40);
    return 0;
}
```

Wskaźniki

1. Stwórz zmienną wskaźnikową przechowującą adres:
 - a. liczby typu `int`,
 - b. liczby typu `double`,
 - c. obiektu typu `string`,
 - d. tablicy liczb typu `int`.
2. Stwórz zmienną `x` typu `int` a następnie wskaźnik przechowujący jej adres. Wykorzystaj wskaźnik do zmiany wartości zmiennej `x`.

Czy podobne zadanie można wykonać na zmiennych innych typów?
3. Jakie wartości po wykonaniu poniższych fragmentów kodów będą posiadać lub na jakie wskazywać zmienne `a`, `b`, `*c`, `*d`

a. `int a, b;`
`int *c, *d;`
`c = &a;`
`a = 3;`
`d = c;`
`b = 6;`

b. `int a, b;`
`int *c, *d;`
`c = &a;`
`a = 3;`
`d = c;`
`b = 6;`
`*c = 4;`

c. `int a, b;`
`int *c, *d;`
`c = &a;`
`a = 3;`
`d = &b;`
`b = 6;`

4. Napisz funkcję porządkującą wartości dwóch zmiennych typu `int`, `a` i `b`, tak by w zmiennej `a` była wartość większa lub równa wartości w zmiennej `b`. Funkcję należy wywołać poprzez instrukcję `f(&a,&b);`.

Przykład:

```
int a = 3;
int b = 6;
```

```
f(&a, &b);
cout << a << endl; // na ekranie pojawi się liczba 6
cout << b << endl; // na ekranie pojawi się liczba 3
```

5. Pamiętając o tym, że nazwa tablicy jest wskaźnikiem do jej pierwszego elementu użyj postinkrementacji (np. `tab++`) do wyświetlenia wszystkich elementów tablicy *tab*.

6. Stwórz wskaźnik do pierwszego elementu tablicy `int tab[10]` i wykorzystaj go do wypełnienia tej tablicy (pseudo)losowymi wartościami z zakresu `<14;55>`. Następnie wykorzystaj ten sam wskaźnik do wyświetlenia wartości tablicy *tab* w odwrotnej kolejności.

7. Stwórz wskaźnik do pierwszego elementu tablicy `int tab[10][10]`. Korzystając z tego wskaźnika wypełnij ją (pseudo)losowymi wartościami z zakresu `<101; 303>`. Użyj utworzony wskaźnik do wyświetlenia wartości tablicy *tab*. Nie korzystaj w tym zadaniu z nawiasów kwadratowych do przemieszczania się po elementach tablicy.

8. Co wypisze program na ekran w wyniku wykonania poniższego fragmentu kodu:

```
int i = 1024; // obiekt typu int
int *p = &i; // wskaźnik do obiektu typu int
int &r = i; // referencja do obiektu typu int
r = 13;
cout << r << endl;
cout << i << endl;
```

Struktury, klasy, unie

1. Opisz poniższy program:
 - a. jakie zadanie realizuje,
 - b. czym jest "a" w funkcji main,
 - c. czy this->a i a to dwie różne zmienne,
 - d. jak nazywa się metodę abc(int),
 - e. dlaczego funkcja abc(int) nie posiada typu,
 - f. co zostanie wyświetlone na ekranie gdy zamiast 1 w instrukcji abc a(1); użyjemy innych wartości całkowitych,
 - g. jak zmodyfikować program by zamiast wartości liczbowych można było używać liter w instrukcji abc a(1); (tworzenie obiektu),
 - h. jak należy zmodyfikować program by realizował założenia szyfru Cezara (wyświetlał odpowiednio zaszyfrowaną literę),
 - i. co zmieni zmiana instrukcji this->a = a+65; na this->a = a%26+65; ,
 - j. jak nazywa się metodę ~abc() w odniesieniu do struktury lub/i klasy.

```
#include <iostream>

using namespace std;

struct abc{
    int a;
    abc(int a)
    {
        this->a = a+65;
    }
    ~abc(){
        cout << "\n zegnam";
    }
    void abc_1()
    {
        cout<< (char)this->a;
    }
    void abc_2();
};

void abc::abc_2(){
    cout << (this->a+3)%26;
}
```

```
int main()
{
```

```

{
    abc a(1);
    a.abc_1();
}
abc *pA;
pA = new abc(4);
pA->abc_1();
pA->abc_2();
return 0;
}

```

2. Sprawdzić wielkość struktury, funkcją `sizeof()`, gdy struktura ma

- a. trzy atrybuty typu `int`, `double`, `char`,
- b. tablicę `double t[100]`,
- c. dwie funkcje `double f1()`, `double f2()`,
- d. dwa parametry typu `int`, `int`, jedną funkcję `void f(int)`.

(źródło: [Programowanie2-cpp-Wyklad-03-ListaZadan.pdf](#))

3. Napisz strukturę