

## Instrukcje pętli -ćwiczenia

### Instrukcja while

Pętla `while` (dopóki) powoduje powtarzanie zawartej w niej sekwencji instrukcji tak długo, jak długo zaczynające pętlę wyrażenie pozostaje prawdziwe.

```
while ( wyrażenie logiczne )
    instrukcja;
```

`wyrażenie logiczne` - jest wyrażeniem języka C++, zaś `instrukcja` - jest dowolną instrukcją lub blokiem instrukcji C++ (ujętym w nawiasy klamrowe). Gdy wartością wyrażenia jest `true` (prawda), wykonywana jest `instrukcja`, po czym następuje powrót do początku pętli i ponowne sprawdzenie wartości wyrażenia. Czynność ta powtarza się, dopóki `wyrażenie logiczne` zwraca wartość `true`. Gdy wyrażenie ma wartość `false`, działanie pętli `while` kończy się i program przechodzi do instrukcji następujących po pętli.

### Instrukcja do...while

Pętla `do...while` (wykonuj...dopóki) wykonuje ciało pętli przed sprawdzeniem warunku i sprawia że instrukcje w pętli zostaną wykonane co najmniej raz.

```
do
    instrukcja
while (wyrażenie);
```

Wykonywana jest instrukcja, po czym sprawdzana jest wartość wyrażenia. Jeśli wyrażenie ma wartość `true`, pętla jest powtarzana; w przeciwnym razie jej działanie się kończy. Pod innymi względami instrukcje i warunki są identyczne, jak w pętli `while`.

### PĘTLA while i do...while: pętla zaporowa.

1. Napisz program wczytujący z klawiatury liczbę całkowitą. Program powinien przyjmować jedynie liczbę dodatnią. Jeśli użytkownik poda liczbę ujemną, to powinien zostać poinformowany, że wymagana jest liczba dodatnia i poproszony o kolejną liczbę. Próbę wczytywania liczby powtarzamy dopóty, dopóki użytkownik nie poda liczby poprawnej (dodatniej). Taki sposób zapewnienia poprawności wczytywanych danych nazywać będziemy *pętlą zaporową*.

*//z użyciem pętli while*

---

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cout << "Podaj liczbę dodatnią";
    cin >> n;
    while (n<=0)
    {
        cout << "Podałeś liczbę ujemną!\n\nPodaj liczbę dodatnią ";
        cin >> n;
    }
    cout << "Liczba dodatnia wynosi:" << n << endl;
    system("PAUSE");
    return 0;
}
```

*//z użyciem pętli do...while*

---

```
#include <iostream>
using namespace std;

int main()
```

```

{
    int n;

    do
    {
        cout << "Podaj liczbę dodatnią";
        cin >> n;
    }while (n<=0);

    cout << "Liczba dodatnia wynosi:" << n << endl;
    system("PAUSE");
    return 0;
}

```

2. Wykorzystując pętlę zaporową napisz program, wyznaczający pole kwadratu. Program jako dane (długość boku) powinien przyjmować wyłącznie liczby dodatnie.
3. Napisz program wyznaczający wartość pierwiastka kwadratowego z liczby. Wprowadzane dane zabezpiecz pętlą zaporową.
4. Napisz program, który wczytuje z klawiatury liczbę, aż do chwili gdy będzie ona należała do przedziału  $\langle 0, 100 \rangle$ .
5. Napisz instrukcje, które ponawiają wczytywanie liczby całkowitej  $x$ , aż do momentu, gdy  $x \in (0, 15]$ .
6. Napisz program, który wczytuje z klawiatury poprawny numer miesiąca, tzn. liczbę z przedziału  $\langle 1, 12 \rangle$ . Zakładamy, że możliwe są tylko 3 próby podania poprawnego numeru.
7. Napisz instrukcję iteracyjną, która pobiera od użytkownika znak  $zn$  typu `char` do momentu, gdy jest on literą 't', 'T', 'n', lub 'N'.

#### **PĘTLA WHILE: inne zastosowania- rozkład liczby**

Napisz program, który

8. dla wczytanej z klawiatury liczby całkowitej wypisuje sumę cyfr tej liczby.

---

```

#include <iostream>
using namespace std;

int main()
{
    int n;
    cout << "Podaj liczbę całkowitą";
    cin >> n;
    int rob=abs(n); //w rob przechowujemy wartość bezwzględną z n
    //(ponieważ liczba mogłaby być ujemna, a suma cyfr jest liczbą dodatnią)
    int s=0; //w zmiennej s będziemy przechowywać sumę liczb
    while(rob > 0) //dopóki są jeszcze cyfry
    {
        s += rob % 10; //do sumy dodajemy ostatnią cyfrę liczby rob
        rob /= 10; //liczbę rob dzielimy przez 10, skracamy w ten sposób ostatnią cyfrę
    }
    cout << "Suma cyfr liczby " << n << "wynosi" << s << endl;

    system("PAUSE");
    return 0;
}

```

---

9. dla wczytanej z klawiatury liczby naturalnej wypisuje ilość cyfr tej liczby.
10. dla podanej liczby całkowitej nieujemnej  $n$ , obliczy sumę wszystkich jej cyfr i wyświetli otrzymany wynik, a następnie powtórzy te dwie czynności dla obliczonej sumy, itd. Ten proces winien być zakończony, gdy obliczona suma cyfr będzie liczbą jednocyfrową, np. dla  $n=7895$ , należy wyświetlić liczby: 29, 11, 2.
11. dla wczytanej z klawiatury liczby naturalnej wypisuje sumę nieparzystych cyfr tej liczby.
12. wczytuje z klawiatury liczbę całkowitą i wyświetla ją w odwrotnej kolejności.

13. sprawdza, czy wczytana z klawiatury liczba jest palindromem, tzn. czytana od końca jest taka sama np. 12321, 234432, 3445.
14. znajduje ilość jedynek w dwójkowym rozwinięciu podanej przez użytkownika liczby naturalnej  $n$ .
15. wczytuje z klawiatury liczbę całkowitą i oblicza iloczyn niezerowych cyfr tej liczby.
16. wczytuje z klawiatury liczbę całkowitą i wyświetla informację, czy wśród cyfr tej liczby jest 1.
17. sprawdza czy więcej jest cyfr parzystych, czy nieparzystych we wczytanej liczbie.
18. dla dodatniej liczby całkowitej  $n$ , podanej przez użytkownika, obliczy ile razy w jej zapisie wystąpiła jej maksymalna cyfra. Przykładowo dla liczby 318338 funkcja powinna zwrócić wartość 2
19. drukuje na ekranie zadaną liczbę zastępując jej kolejne cyfry ich kwadratami (dla liczby 127 drukujemy 1449).
20. drukuje na ekranie zadaną liczbę zastępując jej kolejne cyfry liczbami powstałymi przez dodanie do cyfr liczby 1 ( $1598 = 26109$ ).
21. dla wczytanej z klawiatury liczby naturalnej  $n$  wypisuje jej dzielniki. Zwróć uwagę do jakiej liczby trzeba przeszukiwać liczby będące potencjalnymi dzielnikami.
22. dla wczytanej z klawiatury liczby naturalnej  $n$  sprawdza, czy liczba ta jest pierwsza.
23. dla wczytanej z klawiatury liczby naturalnej wypisuje jej rozkład na czynniki pierwsze. Zorganizuj wyświetlanie tak, by program jako wynik działania wypisywał linijkę postaci np.  
120=2\*2\*2\*3\*5, 35=5\*7
24. dla danej z klawiatury liczby rzeczywistej  $a$  wyznacza najmniejszą liczbę  $n$ , taką że  
 $1+1/2+1/3+\dots+1/n > a$ .

#### **PĘTLA WHILE: pętla z licznikiem**

Napisz program, który

25. oblicza sumę  $1/1+1/2+1/3+\dots+1/n$ , gdzie  $n$  jest podane przez użytkownika.

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n; //n ma być liczbą >0
    do
    {
        cout << "Podaj liczbę całkowitą";
        cin >> n;
    }while (n<=0);
    float s = 0.0f; //suma odwrotności jest liczbą rzeczywistą
    int k=1; //dodatkowy licznik, który będzie przechodził
        // po kolejnych mianownikach od 1 do n
    while(k <= n) //dopóki są jeszcze cyfry
    {
        s += 1.0f/k; //do sumy dodajemy kolejne składniki (1.0- aby dzielenie było
            //rzeczywiste, a nie całkowite)
        k++; //zwiększamy licznik
    }
    cout << "Suma odwrotności wynosi" << s << endl;

    system("PAUSE");
    return 0;
}
```

26. wyznaczy i wyświetli na ekranie sumę liczb naturalnych mniejszych od  $n$  (liczba  $>0$  podana przez użytkownika) zakończonych liczbą 5 lub 19.
27. najpierw wczytuje liczbę naturalną  $n$ , następnie wczytuje  $n$  liczb rzeczywistych i jako wynik działania wyświetla na ekranie liczbę najmniejszą i największą w podanym ciągu.
28. dla wczytanej z klawiatury liczby  $n$  wypisuje na ekranie wszystkie liczby z przedziału  $1 \dots n$  podzielne przez 7 których ostatnia cyfra to 1
29. wczytuje z klawiatury liczby rzeczywiste aż do momentu, gdy podana liczba jest równa zero, a następnie wyświetla na ekranie sumę i średnią arytmetyczną tych liczb.

## Instrukcja `for`

Pętla `for` (dla) łączy trzy etapy w jedną instrukcję. Są to: inicjalizacja, test i inkrementacja. Pierwsza instrukcja służy do inicjalizacji. Można w niej umieścić każdą poprawną instrukcję języka C++, ale zwykle po prostu tworzy się i inicjalizuje zmienną licznikową. Drugą instrukcją jest `wyrażenie`, którym może być każde poprawne wyrażenie języka. Pełni ono taką samą funkcję, jak wyrażenie w pętli `while`. Trzecia instrukcja jest działaniem. Zwykle w jego wyniku wartość zmiennej licznikowej jest zwiększana lub zmniejszana, ale oczywiście można tu zastosować każdą poprawną instrukcję języka C++.

```
for (inicjalizacja; wyrażenie; instr1 )  
    instrukcja;
```

Instrukcja `inicjalizacja` jest używana w celu zainicjalizowania stanu licznika lub innego przygotowania do wykonania pętli. Instrukcja `wyrażenie` jest dowolnym wyrażeniem języka C++, które jest obliczane przed każdym wykonaniem zawartości pętli. Jeśli wyrażenie ma wartość `true`, wykonywane jest ciało pętli, po czym wykonywana jest instrukcja `instr1` z nagłówka pętli (zwykle po prostu następuje inkrementacja zmiennej licznikowej).

### PĘTLA FOR:

30. Wykonaj zadania 26–28 z użyciem pętli `for`.

Napisz program, który

31. dla wczytanej z klawiatury liczby  $n$  wypisze na ekranie liczby  $1, 4, 9, \dots, n^2$ .

---

```
#include <iostream>  
using namespace std;  
  
int main(int argc, char *argv[])  
{  
    int n;  
  
    do  
    {  
        cout << "Podaj liczbę liczb n=";  
        cin >> n;  
    }while(n <= 0);  
    for(int k = 1; k <= n; k++)  
        cout<<k*k<<endl;  
    system("PAUSE");  
    return 0;  
}
```

---

32. pobiera od użytkownika liczbę całkowitą  $n$  i oblicza wartość  $n!$ .

33. wczytuje  $n$  liczb i oblicza sumę ich odwrotności.

34. dla danych z klawiatury liczby rzeczywistej  $a$  i liczby całkowitej  $n$  oblicz  $a^n$

35. wyznacza sumę  $n$  początkowych liczb parzystych. Liczbę  $n$  należy pobrać od użytkownika.

36. wyznacza sumę  $n$  początkowych liczb nieparzystych. Liczbę  $n$  należy pobrać od użytkownika.

37. wyznacza sumę  $n$  ułamków postaci  $1/2, 2/3, 3/4, 4/5, 5/6$  itd.

38. wyznacza sumę  $n$  początkowych liczb podzielnych przez 7. Liczbę  $n$  należy pobrać od użytkownika.

39. liczy sumę  $n$  początkowych liczb naturalnych podzielnych przez 5.

40. liczy sumę  $n$  początkowych liczb naturalnych kończących się jedną z cyfr: 2, 3 lub 9.

41. liczy sumę  $n$  początkowych liczb kończących się jedną z liczb: 1, 5, 11 lub 37.

42. przybliża wartość  $\ln 2$  za pomocą  $n$  początkowych wyrazów szeregu

$$\ln 2 = 1 - 1/2 + 1/3 - 1/4 + \dots + (-1)^{n-1} 1/n$$

43. liczy sumę  $n$  początkowych liczb nieparzystych.

44. drukuje na ekranie  $n$  gwiazdek.

45. drukuje na ekranie liczby. Ilość liczb wczytujemy z klawiatury. Oto wydruk dla  $ile = 5$ :  $0, 3, 6, 9, 12$

## PĘTLA FOR: pętla w pętli - rysowanie wzorków.

Napisz program, który

46. drukuje na ekranie prostokąt z literek X. Wysokość i szerokość prostokąta wczytujemy z klawiatury.

Poniższy prostokąt ma wymiary: szer=10, wys=4.

```
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

---

```
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int szerokosc,wysokosc;

    cout << "SZEROKOSC=";
    cin >> szerokosc;

    cout << "WYSOKOSC=";
    cin >> wysokosc;

    cout << endl;

    for(int i=1; i <= wysokosc; i++)
    {
        for (int j=1; j <= szerokosc; j++)
            cout << "X"; //wyświetlamy 1..szerokość X-ów w i-tym wierszu
        cout << endl; //na koniec i-tego wiersza przechodzimy do nowej linii
    }

    cout << endl;
    system("PAUSE");
    return 0;
}
```

---

47. wyświetli na ekranie tabliczkę mnożenia wymiaru  $n \times n$ ,  $n < 15$ . Zadbaj o nagłówki.

48. dla danej całkowitej liczby  $n$  wyświetli fragment tabliczki mnożenia, jak na przykładowym wydruku dla  $n=5$ .

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

49. drukuje na ekranie kwadrat. Długość boku kwadratu wczytujemy z klawiatury. Poniższy kwadrat ma bok długości 4.

```
KKKK
KKKK
KKKK
KKKK
```

50. drukuje na ekranie liczby. Wysokość wczytujemy z klawiatury. Oto wydruk dla wysokości  $h=5$ :

```
1
2 4
3 6 9
4 8 12 14
5 10 15 20 25
```

51. drukuje na ekranie liczby. Wysokość wczytujemy z klawiatury. Oto wydruk dla wysokości  $h=5$ :

```
0
3 6
6 9 12
9 12 15 18
12 15 18 21 24
```

52. wyświetli na ekranie następujące ciągi liczb

```
a)          b)
1           1 2 3 4 ... n-1 n
1 2        1 2 3 4 ... n-1
1 2 3      .....
1 2 3 4    1 2 3
.....     1 2
1 2 3 4 ... n 1
```

53. wyświetli na ekranie tabelę:

```
2  4  6  8  10
3  6  9  12 15
4  8  12 16 20
5  10 15 20 25
6  12 18 24 30
7  14 21 28 35
```

54. wyświetli na ekranie

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
```

55. wyświetli na ekranie (Dla a) i b) c) wysokość trójkąta wczytujemy z klawiatury. Poniższe trójkąty mają wysokość  $h=4$ .(Dla d) i e) wczytujemy wysokość-  $h$  i szerokość-  $w$ . Poniższe trapezy mają wysokość  $h=4$  i szerokość  $w=3$  )

```
a)          b)          c)          d)          e)
*           *           *           ***          *
**          **          **          ****         *
***         ***         ***         *****      *
****        ****        ****        *
*****      *****    *****    *****      *
```