



Dev-C++

## Część 4 pętla FOR.

### Ćwiczenia wykonane pisemnie (notatki w zeszycie)

Zapisz w zeszycie odpowiedzi na następujące pytania. Przepisz najpierw pytanie **podkreśl na zielono** następnie udziel odpowiedzi.

1. podaj postać pętli for dla jednej instrukcji wykonywanej przez pętlę,
2. podaj postać pętli dla wielu instrukcji wykonywanych przez pętlę,
3. podaj i narysuj za pomocą algorytmu przebieg wykonania pętli for,
4. zapisz przykłady instrukcji for
  - pełna instrukcja for
  - z wyrażeniem inicjującym obliczonym poza pętlą
  - z obliczanym warunkiem przerywania pętlę
5. kiedy pętla nie będzie wykonana
6. napisz na podstawie przykładu 19 instrukcję, która wydaje dźwięk

### Instrukcja pętli iteracyjnej FOR

Postać pętli for jest następująca:

- a) dla jednej instrukcji wykonywanej przez pętlę  
 for (wyrażenie\_inicjujące; wyrażenie\_logiczne; wyrażenie\_kroku)  
     Instrukcja;

przykład

- a) pełna instrukcja for  
 for (int n=0; n<=100; n++) printf("%d\t", n);

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	Aby kontynuować, naciśnij dowolny klawisz . . .								

- b) dla wielu instrukcji wykonywanej przez pętlę  
 for (wyrażenie\_inicjujące; wyrażenie\_logiczne; wyrażenie\_kroku)  
   {  
     Instrukcja1;  
     Instrukcja2;  
     .....  
     Instrukcja;  
   }

**Wykonanie pętli for przebiega następująco:**

1. wykonanie jeden raz wyrażenia inicjującego.
2. obliczenie wartości logicznej wyrażenia logicznego.
3. jeśli w\_logiczne ma wartość prawda (true) nastąpi wykonanie instrukcji.
4. obliczenie wyrażenia kroku.
5. powtórne sprawdzenie warunku - czy wyrażenie logiczne ma wartość różną od zera. jeśli wyrażenie logiczne ma wartość zero, nastąpi zakończenie pętli.

przykład

b)z wyrażeniem inicjującym obliczonym poza pętlą

```
.....
{
float n, p=1;
n=(p+1)/(p+1);
for (; n<=100; n++) printf("%.4f\t", n);
}
```

1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000
11.0000	12.0000	13.0000	14.0000	15.0000	16.0000	17.0000	18.0000	19.0000	20.0000
21.0000	22.0000	23.0000	24.0000	25.0000	26.0000	27.0000	28.0000	29.0000	30.0000
31.0000	32.0000	33.0000	34.0000	35.0000	36.0000	37.0000	38.0000	39.0000	40.0000
41.0000	42.0000	43.0000	44.0000	45.0000	46.0000	47.0000	48.0000	49.0000	50.0000
51.0000	52.0000	53.0000	54.0000	55.0000	56.0000	57.0000	58.0000	59.0000	60.0000
61.0000	62.0000	63.0000	64.0000	65.0000	66.0000	67.0000	68.0000	69.0000	70.0000
71.0000	72.0000	73.0000	74.0000	75.0000	76.0000	77.0000	78.0000	79.0000	80.0000
81.0000	82.0000	83.0000	84.0000	85.0000	86.0000	87.0000	88.0000	89.0000	90.0000
91.0000	92.0000	93.0000	94.0000	95.0000	96.0000	97.0000	98.0000	99.0000	100.0000

Aby kontynuować, naciśnij dowolny klawisz . . .

c)z obliczanym warunkiem przerwania pętli

```
{
float y=0, n=0;
for (; (sqrt(n)-y)<=3.0; n++)
{
    y=sqrt(n);
    printf("%.2.3f\t", y);
}
}
```

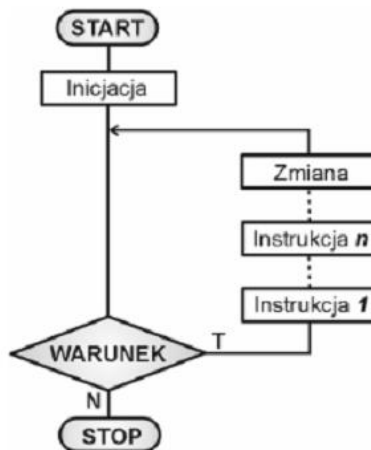
**Uwagi:**Uwaga1

Każde z tych wyrażen może zostać pominięte patrz →for(;;).

Uwaga2

warunek jest testowany przed wykonaniem instrukcji. Jeśli zatem nie zostanie spełniony warunek, instrukcja może nie wykonać się ani razu.

## Projektowanie algorytmów z wykorzystaniem pętli *for*



### Przykład 19

Temat: Program piszący na ekranie 100 wykrzykników w jednym ciągu.

```

#include <cstdlib>
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int main(int argc, char *argv[])
{
    for ( int i=0;i<=99;i++)
    {
        printf("!");
    }
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
  
```

**Uwaga:** Przy rozwiązywaniu następnego zadania wykorzystaj treść tego przykładu.

### Zadanie 30 ( 5pkt)

Temat: Napisać program drukujący 20 gwiazdek. Każda w nowej linii, wykorzystaj instrukcję FOR.



**Uwaga:** Przy rozwiązywaniu następnego zadania wykorzystaj treść tego przykładu.

### **Zadanie 31**

Zmienne z trzema literami nazwiska.

Napisz program z użyciem pętli for wypisujący liczby parzyste od numeru \_miesiąca\_urodzenia (Twojego – nawet jeśli jest on liczbą nieparzystą) do numeru \_miesiąca\_urodzenia +30.

Wykonaj schemat blokowy

**Wskazówka:** w trzeciej sekcji instrukcji for zamiast `i++` w pisz warunek tak, aby zmienna `i` przyjmowała wartości parzyste, czyli liczby zwiększające się o dwa. Uwaga: start pętli: zaczyna się od liczby parzystej a koniec również na liczbie parzystej obu zawartych w przedziale ucznia.

**Zapisz** w zeszycie pod listingiem czcionką 1,5cm, trzecią sekcję pętli (czyli tą, w której było zwiększanie o dwa pętli).

np. dla października 10÷40

```
Podaj miesiąc Twojego urodzenia: 10
10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
Aby kontynuować, naciśnij dowolny klawisz . . .
```

np. dla września 9÷39

```
Podaj miesiąc Twojego urodzenia: 9
10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
Aby kontynuować, naciśnij dowolny klawisz . . .
```

### **Przykład 20**

Temat:

Program sumuje liczby aż do wciśnięcia liczby 0.

W przykładzie poniżej nieskończoną pętlę przerywa po podaniu z klawiatury wartości zera instrukcja `break`.

Wykonaj:

- Wpisz przykład do komputera oraz przetestuj go.
- Wykonaj schemat blokowy przykładu.
- Zapisz w zeszycie pod listingiem czcionka 1,5cm, **nieskończoną pętlę**

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    float a, sigma=0;
    for (;;)          // nieskończona pętla
    {
        printf("\n Podaj liczbe do sumowania\n");
        scanf("%f", &a);
        if (a==0) break;
        sigma+=a;      // inaczej sigma = sigma + a;
        printf("\n SUMA CZESCIOWA: %f",sigma);
    }
    printf("Nastapil BREAK \n");
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Podobnie jak pętla for może być pętlą pustą, możesz również użyć instrukcji pustych w jej nagłówku. Instrukcja for (;;) zawierająca w nagłówku trzy instrukcje puste jest nazywana pętlą **wieczną** lub **nieskończoną**, ponieważ będzie działać dopóki nie napotka w swoim ciele instrukcji break lub nie odłączy Ci prądu.

Instrukcje **continue** i **break** pozwalają zmieniać przebieg wykonania ciała pętli.

Użycie instrukcji continue powoduje skok do początku pętli natomiast break - wyjście z pętli. (dotyczy to zarówno pętli for jak i pętli while, którą poznasz już wkrótce).

Prześledźmy działanie tych instrukcji na naszym roboczym przykładzie.

Po napotkaniu instrukcji `continue`, program przeskakuje do kolejnego wykonania pętli, pomijając pozostałe instrukcje zawarte w ciele funkcji

Po napotkaniu instrukcji `break`, program opuszcza pętlę

```

3  /* run this program using the console window
4
5  int main(int argc, char** argv) {
6      int y=0;
7      for(int i=1; i=3; i++)
8      {
9          y++;
10         if(x==1)
11             continue;
12         if(x==2)
13             break;
14         cout<<x<<endl;
15     }
16     cout<<"koniec petli"<<endl;
17
18     return 0;
19 }

```

### Zadanie 32

Zmienne z trzema literami nazwiska.

Program z przykładu 20 zmodyfikujemy w taki sposób, aby:

- jeśli podamy liczbę dodatnią – program będzie dodawał ją do sumy sigma;
- jeśli podana liczba będzie ujemna – program nie będzie robił nic, pomijał bieżącą pętlę przy pomocy rozkazu **continue**; (Ponieważ warunek wejściowy pętli jest zawsze spełniony, to pętlę zawsze uda się kontynuować.)
- jeśli podamy liczbę zero – program przerwie pętlę instrukcją **break**.

```

Podaj liczbe do sumowania
2
SUMA CZESCIOWA: 2.000000
Podaj liczbe do sumowania
3
SUMA CZESCIOWA: 5.000000
Podaj liczbe do sumowania
9
Podaj liczbe do sumowania
3
SUMA CZESCIOWA: 8.000000
Podaj liczbe do sumowania
0
Nastapil BREAK
Aby kontynuowac, naciśnij dowolny klawisz . . .

```

**Zadanie 33**

**Temat:** Program wypisujący kody ASCII oraz liczby odpowiadające tym kodom.

**ETAP1**

Napisać program drukujący na ekranie kody ASCII dla znaków o numerach od 33 do 255.

Wydruk powinien mieć postać (przykład jednej linii) np.

kod znaku 33      !

.....  
.....

Użyj pętli. W celu wydrukowania znaku o odpowiadającym mu kodzie użyj rzutowania zmiennych (czyli zmiany typu `int` → `char`)

np. `cout<<(char)i<<"\n";`      gdy `i=65` wyświetli się litera A.

**ETAP2**

Zmodyfikuj program tak aby zatrzymał się jeśli zapisze cały ekran ( np. 20 wierszy) i czekał, aż wcisniemy dowolny klawisz. Po wcisnięciu klawisza będzie drukował nowy ekran kodów.

Do zmiany ekranów użyj instrukcji IF oraz `MOD` → % (sprawdzaj podzielność zmiennej sterującej I użytej w pętli FOR przez 20 z użyciem `MOD (%)` → ponieważ użyjemy dwadzieścia wierszy na ekranie) gdy zmienna I jest podzielna przez 20 to zatrzymanie ekranu i czekanie na naciśnięciu dowolnego klawisza (np. poprzez `system("PAUSE");` czyszczenie ekranu (np. poprzez `system("cls");` i nowe 20 kodów ASCII.

**ETAP3**

Zmodyfikuj program tak aby drukował na ekranie trzy słupki kodów i odpowiadających mu znaków.

```

D:\Andrzej\Praca\Lekcje\CPP\Cwiczenia_C++\Cz_4\Zadania\Zadanie_33\z_33.exe
kod znaku 123 {          kod znaku 124 |          kod znaku 125 }
kod znaku 126 ~          kod znaku 127 □          kod znaku 128 Ç
kod znaku 129 ü          kod znaku 130 é          kod znaku 131 à
kod znaku 132 ä          kod znaku 133 ü          kod znaku 134 ć
kod znaku 135 ç          kod znaku 136 ı          kod znaku 137 è
kod znaku 138 Ö          kod znaku 139 ó          kod znaku 140 î
kod znaku 141 Ž          kod znaku 142 Ä          kod znaku 143 Ć
kod znaku 144 È          kod znaku 145 Ĺ          kod znaku 146 Ÿ
kod znaku 147 ô          kod znaku 148 ö          kod znaku 149 Ľ
kod znaku 150 ı          kod znaku 151 Š          kod znaku 152 š
kod znaku 153 Ö          kod znaku 154 Ü          kod znaku 155 Ť
kod znaku 156 ě          kod znaku 157 Ł          kod znaku 158 ×
kod znaku 159 ě          kod znaku 160 á          kod znaku 161 í
kod znaku 162 ó          kod znaku 163 ú          kod znaku 164 Å
kod znaku 165 ą          kod znaku 166 Ž          kod znaku 167 ž
kod znaku 168 Ě          kod znaku 169 ě          kod znaku 170 ı
kod znaku 171 ž          kod znaku 172 Ć          kod znaku 173 š
kod znaku 174 «          kod znaku 175 »          kod znaku 176 ÷
kod znaku 177 ☒          kod znaku 178 ☒          kod znaku 179 |
kod znaku 180 ı          kod znaku 181 Å          kod znaku 182 Å
Aby kontynuować, naciśnij dowolny klawisz . . . _

```

**Zadanie 34**

Napisać program drukujący liczby od `180+nr_w_dzienniku` do `200+nr_w_dzienniku` oraz ich pierwiastki. Wykorzystaj treść poprzedniego programu. Zastanów się jak liczba będzie początkiem pętli a jaka końcem oraz w jaki sposób uzyskać wygląd zadania jak poniżej. Zwróć uwagę, że na zrzucie ekranu widnieją dwa miejsca po przecinku.

```
Printf("y=%6.2f",y);
```

Jest to przykład matrycy 6 znaków dwa miejsca po przecinku



Gdy chcesz użyć do wyświetlania cout dwa miejsca po przecinku możesz uzyskać cout.precision(2);

Przykład wydruku zadania dla ucznia, który ma numer w dzienniku 20.

SQRT(200)=14.14

.....

.....

SQRT(201)=14.18

```

podaj numer w dzienniku: 25
SQRT(205)=14.32
SQRT(206)=14.35
SQRT(207)=14.39
SQRT(208)=14.42
SQRT(209)=14.46
SQRT(210)=14.49
SQRT(211)=14.53
SQRT(212)=14.56
SQRT(213)=14.59
SQRT(214)=14.63
SQRT(215)=14.66
SQRT(216)=14.70
SQRT(217)=14.73
SQRT(218)=14.76
SQRT(219)=14.80
SQRT(220)=14.83
SQRT(221)=14.87
SQRT(222)=14.90
SQRT(223)=14.93
SQRT(224)=14.97
SQRT(225)=15.00
Aby kontynuować, naciśnij dowolny klawisz . . . _

```

### Zadanie 35

**Temat:** Napisać program obliczający sumę szeregu harmonicznego od wyrazu pierwszego do milionowego.

Ćwiczenia wstępne:

oblicz:

$$suma = \sum_{i=2}^4 i^2$$

$$suma = \sum_{i=1}^3 \frac{i}{i+2}$$

zapisz w postaci sumy:

$$suma = 1 + 8 + 27 + 64$$

**Szereg harmoniczny** – szereg liczbowy postaci

$$\sum_{n=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$$

Czyli wyrazy szeregu harmonicznego powstają następująco:



$$a_n = \frac{1}{n}$$

sumę wyrazów od pierwszego do milionowego można zapisać z użyciem znaku sigmy

$$suma = \sum_{i=1}^{1000000} \frac{1}{i}$$

Wskazówka: Należy w każdym kroku pętli obliczyć wartość wyrazu  $a_n$  i dodać do aktualnie obliczanej sumy. Pamiętaj o wyzerowaniu zmiennej zawierającej sumę przed przystąpieniem do sumowania. We wzorze na  $n$ -ty wyraz ciągu wpisz zamiast 1 wpisz 1.0 aby dokonała się konwersja.

```
suma w 999991 pętli=14.3927
suma w 999992 pętli=14.3927
suma w 999993 pętli=14.3927
suma w 999994 pętli=14.3927
suma w 999995 pętli=14.3927
suma w 999996 pętli=14.3927
suma w 999997 pętli=14.3927
suma w 999998 pętli=14.3927
suma w 999999 pętli=14.3927
suma w 1000000 pętli=14.3927

całkowita suma=14.3927
Aby kontynuować, naciśnij dowolny klawisz . . .
```

### Przykład 21

Temat: Napisz program rysujący linię pionową na środku ekranu składającą się z gwiazdek.

Następnie dopisz pętlę rysującą nową linię poziomą na wysokości 10.

```
#include <cstdlib>
#include <iostream>
#include <windows.h>

using namespace std;

void gotoxy(int x, int y)
{
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}

int main(int argc, char *argv[])
{
    for (int i=0;i<=22;i++)
    {
        gotoxy(40,i);
        cout<<"*"<<"\n";
    }
}
```

```

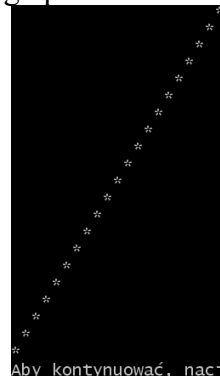
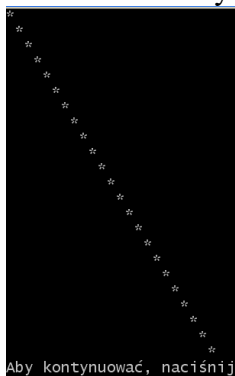
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}

```



### **Zadanie 36**

Zmodyfikuj przykład poprzedni tak aby gwiazdki rysowały się po przekątnej kwadratu o boku 20 linii ekranowych rozpoczynając od lewego górnego punktu ekranu.



### **Zadanie 37**

Zmodyfikuj przykład poprzedni tak aby gwiazdki rysowały się po przekątnej kwadratu o boku 20 linii ekranowych rozpoczynając od prawego górnego punktu ekranu.

### **Zadanie 38**

Napisać program, który z użyciem czterech pętli (instrukcji iteracyjnej FOR) narysuje kwadrat z gwiazdek gdzie:

współrzędna jednego z wierszy to numer wiersza =  $5 + (\text{reszta z dzielenia numer\_z\_dziennika przez } 3)$ , oraz współrzędna jednej z kolumn to numer kolumny =  $11 + (\text{reszta z dzielenia numer\_z\_dziennika przez } 2)$ .

```

Podaj Twój numer w dzienniku: 23
wiersz= 7
kolumna= 12

```

kolumna 12

wiersz 7

```

*****
*      *
*      *
*      *
*      *
*****

```

Aby kontynuować, naciśnij dowolny klawisz . . .

### Zadanie 39

Napisać program, który po wczytaniu środka linii poziomej (poddajesz x i y) oraz długości linii (podajesz D) narysuje ze znaków minus „-„ linię o określony środku 0 (x,y) oraz długości D.

Wygląd ekranu programu:

Podaj zmienną (kolumnę ekranu) x=40

Podaj zmienną (wiersz ekranu) y=10

Podaj długość linii D=11

*I teraz program rysuje tę linię.*

```

Podaj zmienną (kolumnę ekranu)x= 40
Podaj zmienną (wiersz ekranu) y= 10
Podaj długość linii D= 11

```

-----0-----

Aby kontynuować, naciśnij dowolny klawisz . . .

### Przykład 22

Temat: Program do sprawdzania czy liczba wczytana z klawiatury jest liczbą pierwszą.

Wykonaj:

- 1)Wpisz w zeszycie temat programu.
- 2)Przepisz opis problemu.
- 3)Przepisz opis algorytmu.
- 4)Po rozwiązaniu zapisz listing.

Opis problemu:

**Liczba pierwsza**, liczba naturalna  $n > 1$ , dla której istnieją tylko dwa dzielniki naturalne: 1 i n. Największą znaną liczbą pierwszą jest  $2^{69725932}-1$  (7 VII 1999), liczba ta zapisana w systemie dziesiętnym składa się z ponad 2 mln cyfr.

Opis algorytmu sprawdzania czy liczba jest liczbą pierwszą.

Sprawdzamy czy kolejne liczby naturalne od 2 do pierwiastek(n) są dzielnikami liczby n. Sprawdzenie podzielności odbywa się poprzez użycie funkcji % (MOD→reszta z dzielenia), jeśli reszta z dzielenia jest zero to oznacza to, że liczba n jest podzielna przez liczbę mniejszą od n i n nie jest liczbą pierwszą.

Należy zauważyć, że nie jest konieczne sprawdzanie kolejnych liczb naturalnych od 2 do  $n$ , a wystarczy do **pierwiastek( $n$ )**. Zmienna **k** ma wartość domyślnie zero. Po zakończeniu programu sprawdzamy wartość zmiennej **k**, jeśli jest równa zero oznacza to, że liczba  $n$  jest pierwsza, jeśli nie (czyli jeden) to liczba nie jest pierwsza. Zmiana **k** przybiera wartość jeden gdy liczba  $n$  jest podzielna przez kolejną liczbę naturalną mniejszą od **pierwiastka( $n$ )**.

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    cout<<"podaj liczbe naturalna, ktora sprawdzę czy jest pierwsza=";
    int pierwsza;
    int k;
    cin>>pierwsza;
    int i;
    k=0;
    for( i=2;i<=sqrt(pierwsza);++i)
    {
        if ((pierwsza) % i==0)
        {
            k=1;
            cout<<"liczba "<<pierwsza<<" to nie jest liczba pierwsza jest podzielna przez "
            <<i<<"\n";
        }
    }
    if ( k==0)
    {
        cout<<"liczba "<<pierwsza<<" to jest liczba pierwsza"<<"\n";
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

#### **Zadanie 40**

Z użyciem powyższego programu znajdź i zapisz w zeszycie liczbę pierwszą, która jest większą najbliższą lub równą **liczbie=1000+nr\_z\_dziennika\*100+15**

**Np. nr 21**

**Liczba=1000+2100+15=3115**

```
podaj numer z dziennika=21
podana liczba=1000+nr_z_dziennika*100+15 wynosi 3115
najbliższa liczba pierwsza to 3119
Aby kontynuować, naciśnij dowolny klawisz . . . _
```

**Zadanie 41**

Napisać program, który po wczytaniu dwóch liczb naturalnych większych od 1, czyli liczba **dol** oraz liczba **gor** gdzie  $dol < gor$ . Wypisze sumę liczb pierwszych z tego przedziału.

Wskazówka:

Wykorzystaj poprzedni przykład. Utwórz pętlę dla zmiennej **j\_naz** ( z trzema literami nazwiska) od liczby **dol** do liczby **gor**. Utwórz zmienną **suma**. Pamiętaj o wyzerowaniu tej zmiennej. Użyj instrukcji **suma=suma + j\_naz**. Po zakończeniu pętli wyprowadź zmienna **suma**. Przy wczytywaniu danych pamiętaj, że najmniejsza liczba pierwsza to 2.

Np.

Podaj dol=2

Podaj gor=10

Suma=17

```
podaj dolną wartość dol=2
podaj górną wartość gora=10
liczba 2 to jest liczba pierwsza
liczba 3 to jest liczba pierwsza
liczba 5 to jest liczba pierwsza
liczba 7 to jest liczba pierwsza
suma=17
Aby kontynuować, naciśnij dowolny klawisz . . . _
```

**Zadanie 45**

Wyłącznie za pomocą pętli for napisz program drukujący na ekranie tabliczkę mnożenia do 7:

```
=====
| 1 2 3 4 5 6 7
=====
1| 1 2 3 4 5 6 7
2| 2 4 6 8 10 12 14
3| 3 6 9 12 15 18 21
4| 4 8 12 16 20 24 28
5| 5 10 15 20 25 30 35
6| 6 12 18 24 30 36 42
7| 7 14 21 28 35 42 49
Aby kontynuować, naciśnij dowolny klawisz . . .
```

**Przykładowy sprawdzian****Zadanie1(1 punkt)**

Napisać program, który po wczytaniu dwóch liczb naturalnych większych od 1, czyli liczba **dol** oraz liczba **gor** gdzie  $dol < gor$ . Wypisze wszystkie liczby pierwsze bliźniacze. Liczby pierwsze bliźniacze to dwie liczby pierwsze różniące się między sobą o dwa np. 3 i 5.

Wygląd ekranu:

Podaj dolną liczbę przedziału przeszukiwania=1

Podaj górną liczbę przedziału przeszukiwania=20

Znalezione liczby bliźniacze to :

```
3      5
5      7
11     13
17     19
```

```
podaj dolną wartość dol=1
podaj górną wartość gora=20
wartości muszą być większe od 1 oraz wartość gor musi być większa od dol
podaj dolną wartość dol=2
podaj górną wartość gora=20

Znalezione liczby bliźniacze to :
      3      5
      5      7
     11     13
     17     19
Aby kontynuować, naciśnij dowolny klawisz . . . _
```

### Zadanie2 (1 punkt)

Napisać program obliczając:

- sumę dwucyfrowych liczb naturalnych, czyli :

$$suma = \sum_{i=10}^{99} i$$

```
suma w 88 pętli=3871
suma w 89 pętli=3960
suma w 90 pętli=4050
suma w 91 pętli=4141
suma w 92 pętli=4233
suma w 93 pętli=4326
suma w 94 pętli=4420
suma w 95 pętli=4515
suma w 96 pętli=4611
suma w 97 pętli=4708
suma w 98 pętli=4806
suma w 99 pętli=4905

całkowita suma=4905
Aby kontynuować, naciśnij dowolny klawisz . . .
```

- średnią dwucyfrowych liczb naturalnych (podziel sumę przez odpowiednią wartość)

```
pętla 90 suma=4050 srednia=45.00
pętla 91 suma=4141 srednia=45.51
pętla 92 suma=4233 srednia=46.01
pętla 93 suma=4326 srednia=46.52
pętla 94 suma=4420 srednia=47.02
pętla 95 suma=4515 srednia=47.53
pętla 96 suma=4611 srednia=48.03
pętla 97 suma=4708 srednia=48.54
pętla 98 suma=4806 srednia=49.04
pętla 99 suma=4905 srednia=49.55

całkowita suma=4905

średnia=49.55
Aby kontynuować, naciśnij dowolny klawisz . . .
```

- ilość liczb dwucyfrowych podzielnych przez liczbę wczytaną z klawiatury.

Program powinien wypisać:

Suma to:.....

Średnia to:.....

Ilość liczb podzielnych przez..... to .....

Użyj pętli.

```

Podaj liczbę, której podzielność będziesz badał=5
suma to 4905
średnia to 49.55
ilość liczb podzielnych przez 5 to 18
Aby kontynuować, naciśnij dowolny klawisz . . .

```

### Zadanie3(1 punkt)

Wykonaj program rysujący linię pionową składającą się z podwójnej ilości gwiazdek na środku ekranu o wczytanej długości. Długość linii od 2 do 20.

np. dla dlu=5

```

**
**
**
**
**

Podaj długość linii D= 5

**
**
**
**
**

Aby kontynuować, naciśnij dowolny klawisz . . .

```

### Zadanie4(1 punkt)

Narysuj figurę jak poniżej z gwiazdek o długości i wysokości liczba\_liter\_imienia+3 użyciem dwóch pętli. Od miejsca ekranu (x,y) x i y wczytane z klawiatury jest to położenie pierwszej gwiazdki u góry po lewej stronie.

```

*****
*
*
*
*
*
*

```

### Zadanie5 (1 punkt)

Napisać program piszący 20 razy nazwisko ucznia. Każde w nowej linii, i przesunięte dwie kolumny w lewo wykorzystaj instrukcję FOR. Początek pisania taki aby 20 napisów zmieściło się na ekranie.

np.





### **Zadanie6 (1 punkt)**

Napisać program drukujący liczby, ich kwadraty i sześciany od numeru w dzienniku do numeru w dzienniku+15

Przykład wydruku zadania (wygląd musi być identyczny)

I=4	I*I=16	I*I*I=64
I=5	I*I=25	I*I*I=125

.....  
 .....

```

podaj numer w dzienniku: 30
I=30      I*I=900      I*I*I=27000
I=31      I*I=961      I*I*I=29791
I=32      I*I=1024     I*I*I=32768
I=33      I*I=1089     I*I*I=35937
I=34      I*I=1156     I*I*I=39304
I=35      I*I=1225     I*I*I=42875
I=36      I*I=1296     I*I*I=46656
I=37      I*I=1369     I*I*I=50653
I=38      I*I=1444     I*I*I=54872
I=39      I*I=1521     I*I*I=59319
I=40      I*I=1600     I*I*I=64000
I=41      I*I=1681     I*I*I=68921
I=42      I*I=1764     I*I*I=74088
I=43      I*I=1849     I*I*I=79507
I=44      I*I=1936     I*I*I=85184
I=45      I*I=2025     I*I*I=91125
Aby kontynuować, naciśnij dowolny klawisz . . .
  
```

### **ZADANIE 7 (1 punkt)**

Liczby pierwsze w postaci  $p$ ,  $p+2$ ,  $p+6$ ,  $p+8$  nazywa się czworaczkami (np.: 5,7,11,13,17,19 101,103,107,109...). Nie wiemy, czy jest ich skończenie, czy nie skończenie wiele. Napisz program znajdujący czworaczki.

```
podaj dolną wartosc dol=2
podaj górą wartość gora=1000

Znalezione czworaczki to :
      5      7      11      13
     11     13     17     19
    101    103    107    109
    191    193    197    199
    821    823    827    829
Aby kontynuować, naciśnij dowolny klawisz . . .
```