

Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości	
Sprawozdanie nr: 4	Temat: Pętle
Wydział: Informatyki i Nauk o Żywności	Autor: Damian Jakub Kaczyński Nr albumu: 10492
Przedmiot: Podstawy Programowania	Prowadzący: mgr inż. Paweł Kamiński
Data: 14.11.2021	Ocena:
Uwagi:	

Wstęp

Na zajęciach poznaliśmy różne rodzaje pętli i dostaliśmy 13 zadań do wykonania z ich użyciem. Zadania te polegają na zautomatyzowaniu powtarzalnych czynności tak żeby komputer wykonał żmudną, powtarzalną pracę.

Treść zadania 1

Napisz program, który wypisze liczby od 1 do 100.

Rozwiązanie

```
#include <stdio.h>
```

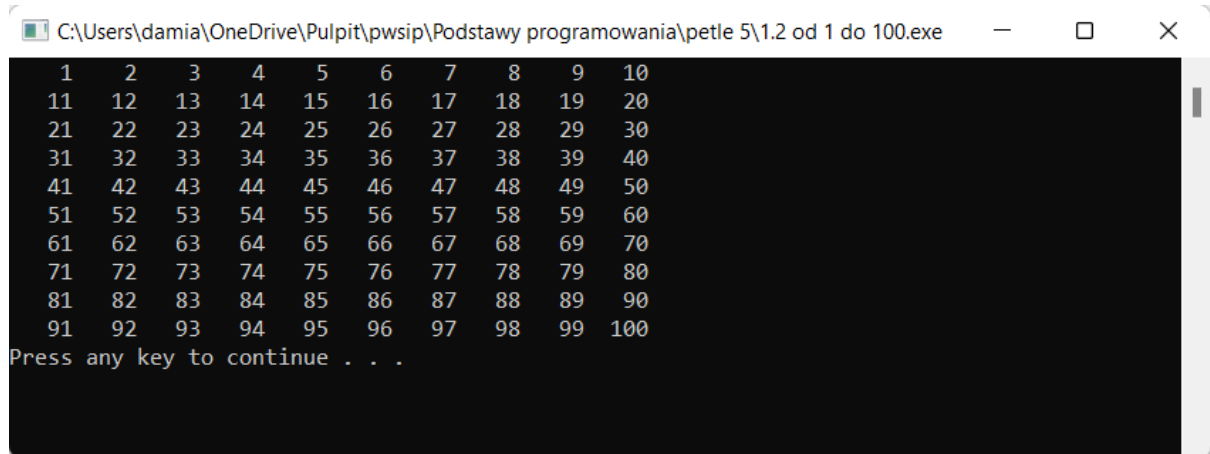
```
#include <stdlib.h>
```

```
int main(){
```

```
    for(int i = 1; i <= 100; i++){
        printf("%5d", i);
        if(i % 10 == 0){
            printf("\n");
        }
    }
}
```

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

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\1.2 od 1 do 100.exe  
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  
Press any key to continue . . .
```

Treść zadania 2

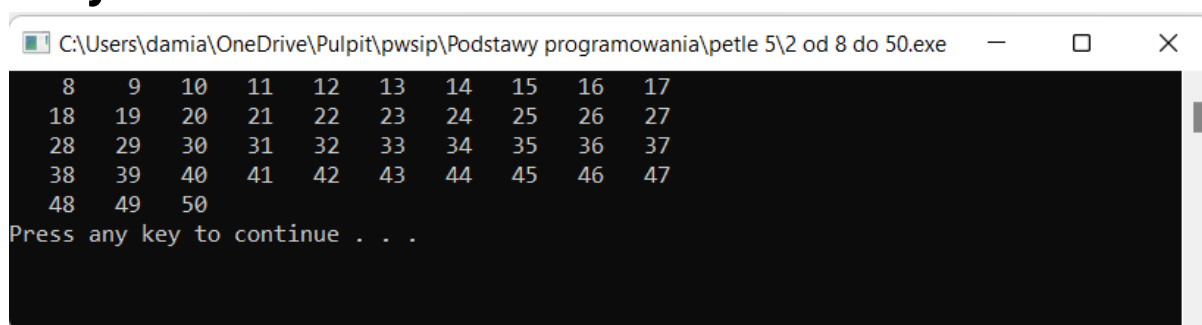
Napisz program, który wypisze liczby od 8 do 50.

Rozwiązanie

```
#include <stdio.h>  
#include <stdlib.h>
```

```
int main(){  
    for(int i = 8, j = 0; i <= 50; i++){  
        printf("%5d", i);  
        j++;  
        if(j % 10 == 0){  
            printf("\n");  
        }  
    }  
    printf("\n");  
    system("pause");  
    return 0;  
}
```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\2 od 8 do 50.exe
 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
Press any key to continue . . .
```

Treść zadania 3

Napisz program, który wypisze liczby parzyste od 2 to 50.

Rozwiązanie

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(){
```

```
    for(int i = 2; i <= 50; i += 2){
```

```
        printf("%5d", i);
```

```
        if(i % 20 == 0){
```

```
            printf("\n");
```

```
        }
```

```
    }
```

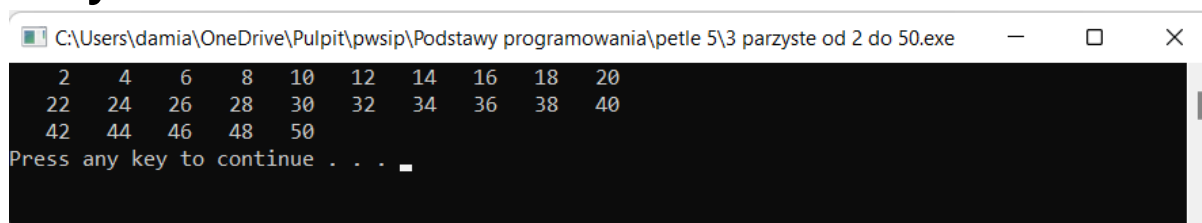
```
    printf("\n");
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\3 parzyste od 2 do 50.exe
 2  4  6  8 10 12 14 16 18 20
22 24 26 28 30 32 34 36 38 40
42 44 46 48 50
Press any key to continue . . .
```

Treść zadania 4

Napisz program, który wypisze liczby od 100 do 1.

Rozwiązanie

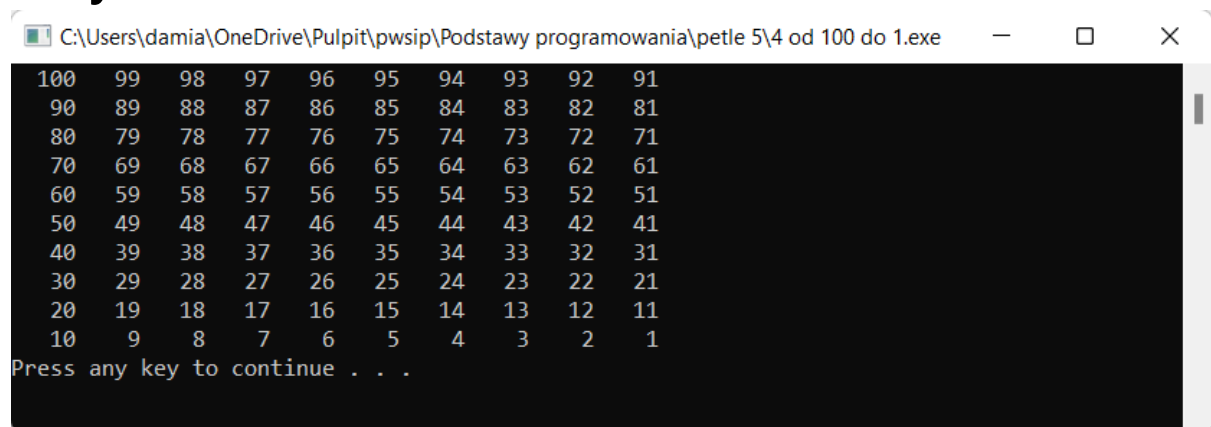
```
#include <stdio.h>
#include <stdlib.h>

int main(){

    for(int i = 100; i >= 1; i--){
        if(i % 10 == 0 && i != 100){
            printf("\n");
        }
        printf("%5d", i);
    }
    printf("\n");

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

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\4 od 100 do 1.exe
100  99  98  97  96  95  94  93  92  91
 90  89  88  87  86  85  84  83  82  81
 80  79  78  77  76  75  74  73  72  71
 70  69  68  67  66  65  64  63  62  61
 60  59  58  57  56  55  54  53  52  51
 50  49  48  47  46  45  44  43  42  41
 40  39  38  37  36  35  34  33  32  31
 30  29  28  27  26  25  24  23  22  21
 20  19  18  17  16  15  14  13  12  11
 10   9   8   7   6   5   4   3   2   1
Press any key to continue . . .
```

Treść zadania 5

Napisz program, który wypisze liczby podzielne przez 8 (w zakresie 1..100).

Rozwiązanie

```
#include <stdio.h>
#include <stdlib.h>

int main(){
```

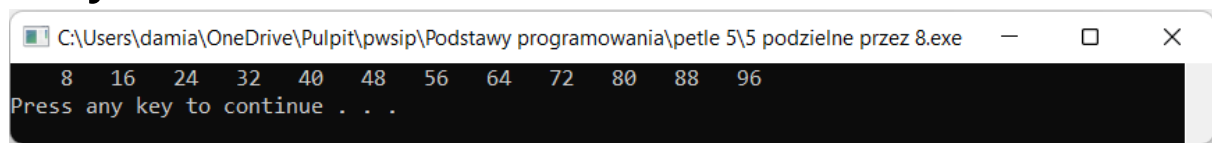
```

    for(int i = 1; i <= 100; i++){
        if(i % 8 == 0){
            printf("%5d", i);
        }
    }
    printf("\n");

    system("pause");
    return 0;
}

```

Przykładowe uruchomienie



Treść zadania 6

Napisz program, który wypisze liczby podzielne przez 3 lub 5 (w zakresie 1..100).

Rozwiązanie

```

#include <stdio.h>
#include <stdlib.h>

```

```

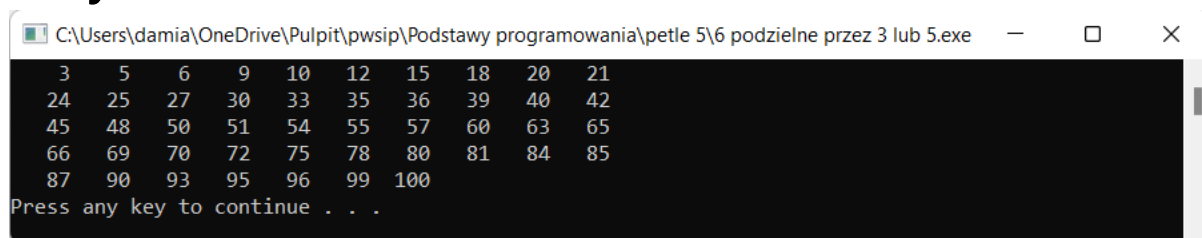
int main(){

    for(int i = 1, j = 0; i <= 100; i++){
        if(i % 3 == 0 || i % 5 == 0){
            j++;
            printf("%5d", i);
            if(j % 10 == 0){
                printf("\n");
            }
        }
    }
    printf("\n");

    system("pause");
    return 0;
}

```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\6 podzielne przez 3 lub 5.exe
3 5 6 9 10 12 15 18 20 21
24 25 27 30 33 35 36 39 40 42
45 48 50 51 54 55 57 60 63 65
66 69 70 72 75 78 80 81 84 85
87 90 93 95 96 99 100
Press any key to continue . . .
```

Treść zadania 7

Napisz program, który wypisuje liczby od 1 do 50, a następnie od 50 do 1.

Rozwiązanie

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(){
```

```
    for(int i = 1; i <= 50; i++){
        printf("%5d", i);
        if(i % 10 == 0){
            printf("\n");
        }
    }
```

```
    printf("\n");
```

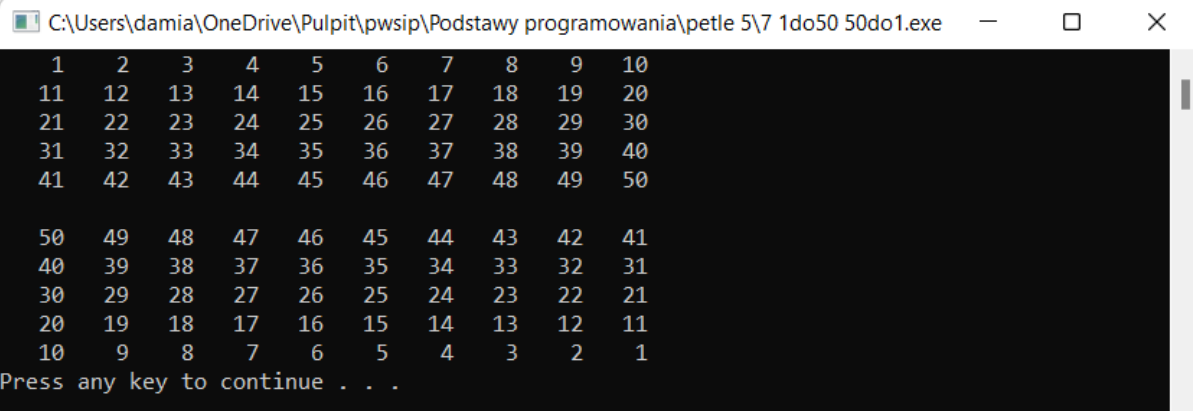
```
    for(int i = 50; i >= 1; i--){
        if(i % 10 == 0 && i != 50){
            printf("\n");
        }
        printf("%5d", i);
    }
```

```
    printf("\n");
```

```
    system("pause");
    return 0;
```

```
}
```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\7 1do50 50do1.exe
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

50 49 48 47 46 45 44 43 42 41
40 39 38 37 36 35 34 33 32 31
30 29 28 27 26 25 24 23 22 21
20 19 18 17 16 15 14 13 12 11
10  9  8  7  6  5  4  3  2  1
Press any key to continue . . .
```

Treść zadania 8

Napisz program wyznaczający sumę liczb podawanych z klawiatury tak długo, aż użytkownik nie wprowadzi liczby jeden.

Rozwiązanie

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int zKlawiatury = 0, wynik = 0;
```

```
    while(zKlawiatury != 1){
```

```
        printf("Podaj liczbe do dodania: \n");
```

```
        scanf("%d", &zKlawiatury);
```

```
        printf("Suma podanych przez ciebie liczby wynosi: %d \n", wynik +=  
zKlawiatury);
```

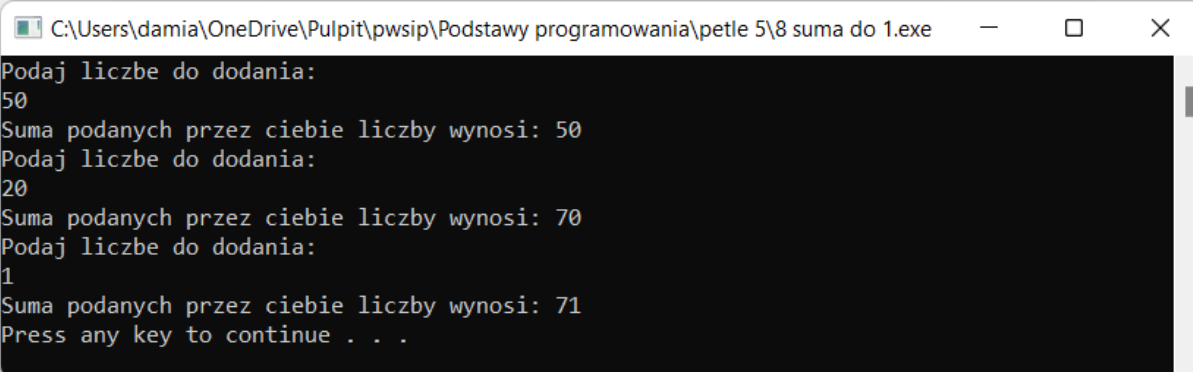
```
    }
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\8 suma do 1.exe
Podaj liczbe do dodania:
50
Suma podanych przez ciebie liczby wynosi: 50
Podaj liczbe do dodania:
20
Suma podanych przez ciebie liczby wynosi: 70
Podaj liczbe do dodania:
1
Suma podanych przez ciebie liczby wynosi: 71
Press any key to continue . . .
```

Treść zadania 9

Napisać program, który będzie wczytywał liczby dopóki ich suma będzie mniejsza od 1000.

Rozwiązanie

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int zKlawiatury, wynik = 0;
```

```
    while(wynik < 1000){
```

```
        printf("Podaj liczbe do dodania: \n");
```

```
        scanf("%d", &zKlawiatury);
```

```
        printf("Suma podanych przez ciebie liczby wynosi: %d \n", wynik +=
```

```
zKlawiatury);
```

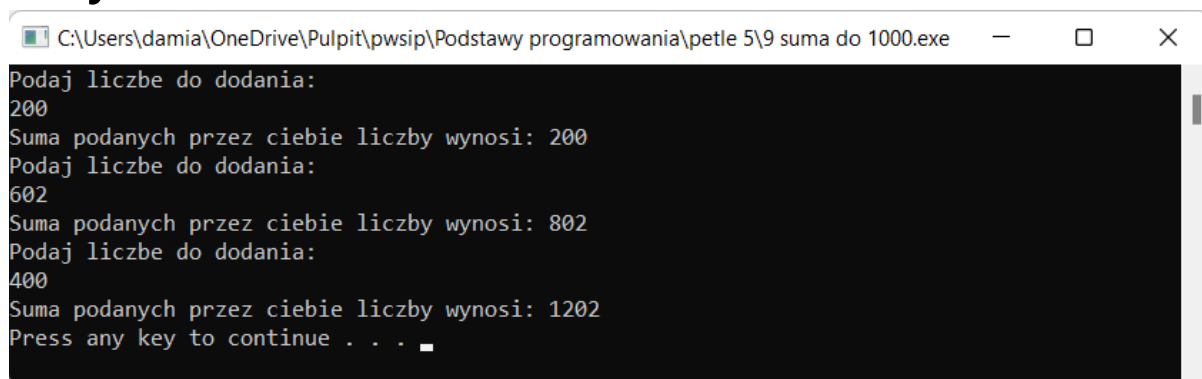
```
    }
```

```
    system("pause");
```

```
    return(0);
```

```
}
```


Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\9 suma do 1000.exe
Podaj liczbe do dodania:
200
Suma podanych przez ciebie liczby wynosi: 200
Podaj liczbe do dodania:
602
Suma podanych przez ciebie liczby wynosi: 802
Podaj liczbe do dodania:
400
Suma podanych przez ciebie liczby wynosi: 1202
Press any key to continue . . .
```

Treść zadania 10

Napisz program drukujący na ekranie tyle gwiazdek, ile podał użytkownik.

Rozwiązanie

```
#include <stdio.h>
#include <stdlib.h>

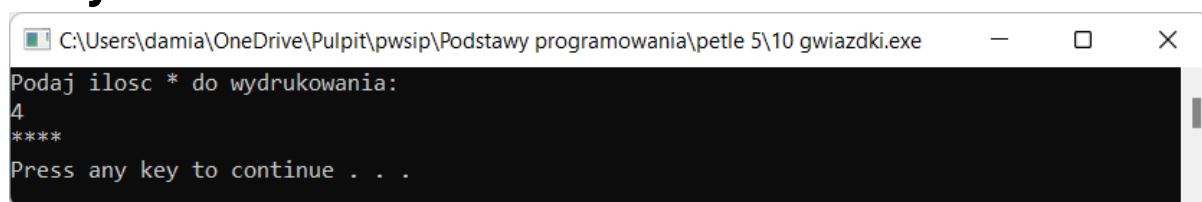
int main() {

    int iloscGwiazdek;
    printf("Podaj ilosc * do wydrukowania: \n");
    scanf("%d", &iloscGwiazdek);

    for(; iloscGwiazdek > 0; iloscGwiazdek--){
        printf("*");
    }
    printf("\n");

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

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\10 gwiazdki.exe
Podaj ilosc * do wydrukowania:
4
****
Press any key to continue . . .
```

Treść zadania 11

Napisz program, który wczytuje z klawiatury n liczb rzeczywistych , a następnie wyświetla na ekranie sumę i średnią arytmetyczną tych liczb.

Rozwiązanie

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int iloscLiczby;
```

```
    float liczba, suma = 0;
```

```
    printf("Podaj ilosc liczb ktore wprowadzisz: \n");
```

```
    scanf("%d", &iloscLiczby);
```

```
    for(int i = 1; i <= iloscLiczby; i++){
```

```
        printf("Wprowadz liczbe %d : \n", i);
```

```
        scanf("%f", &liczba);
```

```
        suma += liczba;
```

```
    }
```

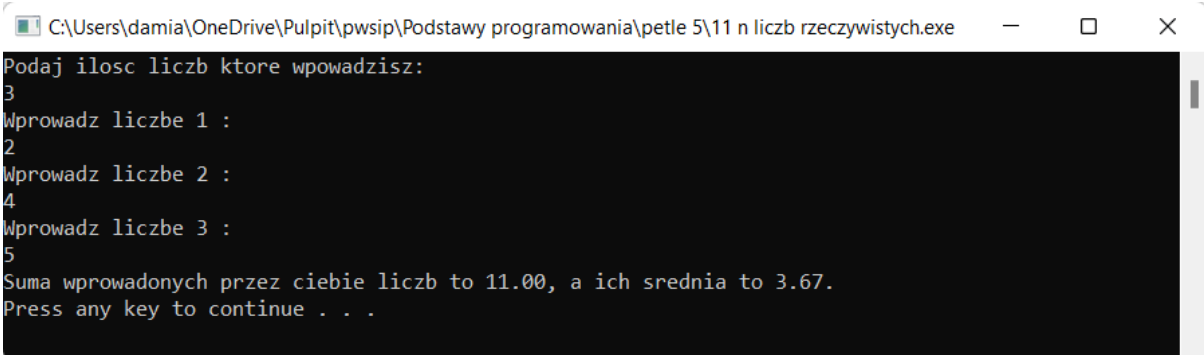
```
    printf("Suma wprowadzonych przez ciebie liczb to %.2f, a ich srednia to %.2f.\n", suma, suma / iloscLiczby);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\11 n liczb rzeczywistych.exe
Podaj ilosc liczb ktore wprowadzisz:
3
Wprowadz liczbe 1 :
2
Wprowadz liczbe 2 :
4
Wprowadz liczbe 3 :
5
Suma wprowadzonych przez ciebie liczb to 11.00, a ich srednia to 3.67.
Press any key to continue . . .
```

Treść zadania 12

Wyznacza sumę n początkowych liczb parzystych. Liczbę n należy pobrać od użytkownika.

Rozwiązanie

```
#include <stdio.h>
#include <stdlib.h>

int main() {

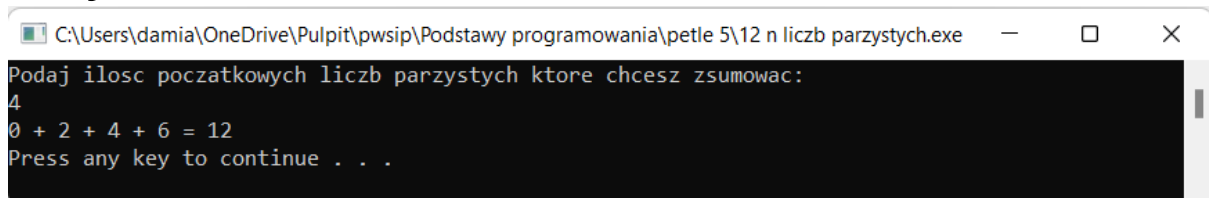
    int iloscLiczby;
    float suma = 0;

    printf("Podaj ilosc początkowych liczb parzystych ktore chcesz zsumowac:
\n");
    scanf("%d", &iloscLiczby);

    for(int i = 0, j = 0; j < iloscLiczby; i++){
        if(i % 2 == 0){
            printf("%d", i);
            suma += i;
            j++;
        }
        if(i % 2 == 0 && j != iloscLiczby){
            printf(" + ");
        }
    }
    printf(" = %.0f \n", suma);

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

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\12 n liczb parzystych.exe
Podaj ilosc początkowych liczb parzystych ktore chcesz zsumowac:
4
0 + 2 + 4 + 6 = 12
Press any key to continue . . .
```

Treść zadania 13

Napisz program, który wyświetli n kolejnych potęg naturalnych liczby 2.

Rozwiązanie

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main() {

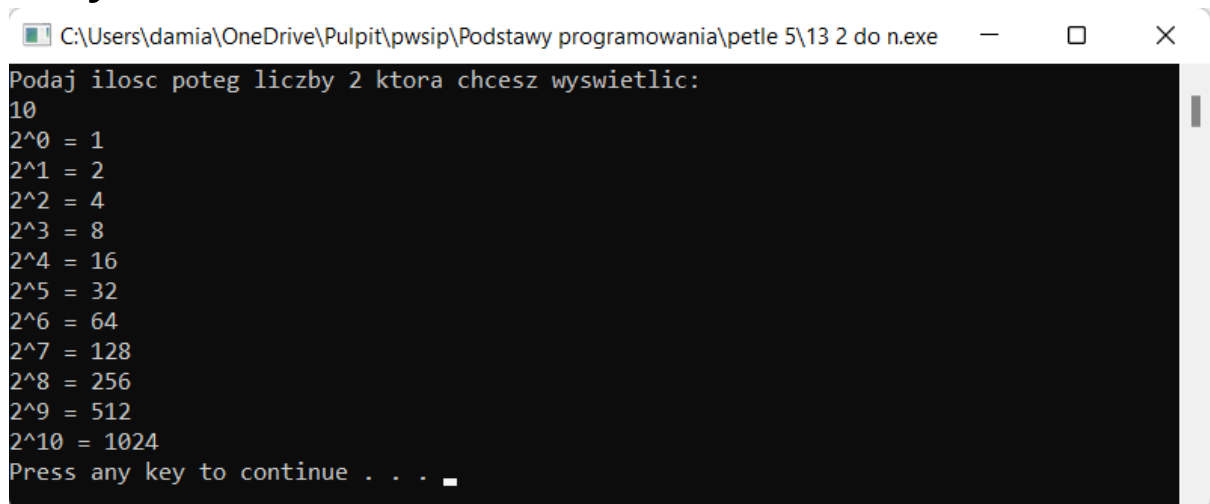
    int iloscPoteg;

    printf("Podaj ilosc poteg liczby 2 ktora chcesz wyswietlic: \n");
    scanf("%d", &iloscPoteg);

    for(int i = 0; i <= iloscPoteg; i++){
        printf("2^%d = %.0f \n", i, pow(2,i));
    }

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

Przykładowe uruchomienie



```
C:\Users\damia\OneDrive\Pulpit\pwsip\Podstawy programowania\petle 5\13 2 do n.exe
Podaj ilosc poteg liczby 2 ktora chcesz wyswietlic:
10
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024
Press any key to continue . . .
```

Wnioski

Użycie pętli pozwoliło na zaoszczędzenie wielu linijek kodu np. w zadaniu pierwszym gdyby pisać do każdej ze stu liczb linię kodu odpowiedzialnej za jej wyświetlenie na ekranie, zajęłoby to ponad 100 linii w porównaniu do 15 linii zawierających całą resztę programu w tym element odpowiedzialny za wyświetlanie po 10 pozycji w każdej linii uruchomionego programu. Użycie pętli w tym przypadku ma jeszcze jedną zaletę: jeśli chcemy zmienić zakres wyświetlanych liczb np na 1 do 200 wystarczy zmienić jedną wartość, nie trzeba dopisywać kolejnych 100 linii kodu.

W trakcie wykonywania zadań zorientowałem się że standardowy sposób wyświetlania wartości jest bardzo nieczytelny. Można było wyświetlić liczby jedna po drugiej oddzielone spacjami lub każdą w nowej linii. Żadnego z tych rozwiązań nie można było nazwać optymalnym, dlatego postanowiłem że w programach w których jest to uzasadnione poświęcę prostotę kodu na rzecz lepszego UX i grupowałem liczby po 10 w linii. Dla poprawienia czytelności programów postanowiłem także dodać elementy informujące użytkownika jak wprowadzane przez niego dane wpływają na przebieg pracy programu.