

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
Кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи № 12
«Програмування, частина 2»

Підготував:
ст. гр. ІХ-11
Диркавець Максим
Перевірив:
Асистент каф РЕПС
Чайковський
І.Б.

Львів 2024

Тема: Оператори циклу

Мета роботи: Ознайомитися з особливостями функціонування операторів циклу та навчитись їх використовувати у процесі програмування.

Хід роботи

- 1) Ознайомитися з теоретичними відомостями.
- 2) Здійснити виконання усіх прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати їх виконання у звіті.

У наступному прикладі в циклі for виводяться на екран числа від 1 до 100:

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

```
int main (void) {
```

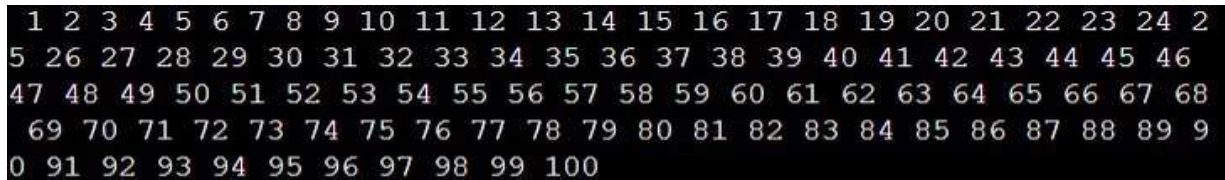
```
int x;
```

```
for (x = 1; x <= 100; x ++)
```

```
printf ( "%d", x);
```

```
return 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
```

Програма для виводу на екран степеня двійки до 1024

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

```
int main() {
```

```
int i=2;
```

```
while (i<=1024)
```

```
{
```

```
    i = i*2;
```

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

```
}
```

```
}
```



```
512
1024
2048
```

Програма демонструє виведення на екран значення числа $k = 50$

```
#include<stdio.h>

int main() {
    int j=0,k=0;
    while(j<5)
    {
        k+=10;
        j++;
    }
    printf("k=%d\n",k);
}
```



```
k=50
```

Наступна програма демонструє виведення на екран значення чисел b, c, a .

```
#include<stdio.h>

int main() {
    int a=2, b=10,c;
    do
    {
        b=b+a; c=10*a; a++;
    }
    while(a<5);
    printf("\nb=%d c=%d a=%d",b,c,a);
}
```

```
b=19 c=40 a=5
```

Нижченаведена програма демонструє процес введення числа до того моменту, поки його значення не буде співпадати з магічним числом.

```
#include<stdio.h>
#include<stdlib.h>
int main(void) {
    int magic;
    int guess;
    magic=rand();
    printf("vgaday chuslo:");
    scanf("%d",&guess);
    if(guess==magic){
        printf("virno");
        printf("magichne chuslo rivne %d",magic);
    }
    else {
        printf("nevirno\n");
        while (guess!=magic){
            printf("vgaday chuslo:");
            scanf("%d",&guess);
            printf("nevirno\n");
        }
    }
}
```

```
vgaday chuslo:2
nevirno
vgaday chuslo:3
nevirno
vgaday chuslo:4
nevirno
vgaday chuslo:124
```

Наступна програма після введення кожного числа здійснює підрахунок кількості спроб та виводить це значення.

```
#include<stdio.h>

#include<stdlib.h>

int main(void) {
    int magic;
    int guess;
    int m=1;
    magic=rand();

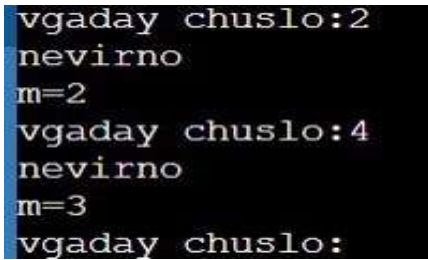
    printf("vgaday chuslo:");
    scanf("%d",&guess);
    if(guess==magic){
        printf("virno");
        printf("magichne chuslo rivne %d",magic);
        }

    else {
        printf("nevirno\n");
        printf("m=%d\n",m);
        while (guess!=magic){
            m++;
        }
        printf("vgaday chuslo:");
        scanf("%d",&guess);
    }
}
```

```

printf("nevirno\n");
printf("m=%d\n",m);
    }
}
}

```



```

vgaday chuslo:2
nevirno
m=2
vgaday chuslo:4
nevirno
m=3
vgaday chuslo:

```

У наступній програмі введено обмеження на максимальну кількість спроб введення числа, яке дорівнює значенню рандомно згенерованого магічного числа. Після використання к-сті заданих спроб, виводиться значення магічного числа.

```

#include <stdio.h>
#include <stdlib.h>
int main(void) {
    int magic;
    int guess;
    int m=1; /*кількість спроб*/
    magic=rand();
    printf("vgaday chuslo:");
    scanf("%d",&guess);
    if(guess==magic){
        printf("virno");
        printf("magichne chuslo rivne %d", magic);
    }
    else {
        printf("nevirno\n");
    }
}

```

```

printf("m=%d\n",m);
while (guess!=magic){
m++;
printf("vgaday chuslo:");
scanf("%d", &guess);
printf("nevirno\n");
printf("m=%d\n",m);
if(m>9){
printf("magic=%d\n", magic);
break;
        }
    }
}

```



```

nevirno
m=10
magic=1804289383

```

3. Виконати нижченаведену програму для обчислення таблиці переведення температури за шкалою Фаренгейта в температуру за шкалою Цельсія. Скрін коду програми та результати її виконання представити у звіті.

```

/* Celsius and Fahrengeit */
/* C=(5/9)*(F-32) */
#include<stdio.h>
#include<conio.h>
int main() {
int fahr, celsius;
int lower, upper, step;

```

```

lower = 0;
upper = 300;
step = 20;
fahr = lower;
printf("\n\nCelsius Fahrenheit\n");
while (fahr <= upper) {
    celsius = 5 * (fahr - 32) / 9;
    printf("%10d\t%8d\n", fahr, celsius);
    fahr = fahr + step;
}
getch();
}

```



Celsius	Fahrenheit
0	-17
20	-6
40	4
60	15
80	26
100	37
120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

4. Скласти програму для створення прямокутного трикутника із зірочок (*), при цьому трикутник має розміри: n рядків у висоту та n символів у ширину. Значення n вводиться з клавіатури. Скрін коду програми та результати її виконання представити у звіті.

```

#include <stdio.h>

int main() {
    int n;
    printf("Введіть розмір трикутника: ");
    scanf("%d", &n);

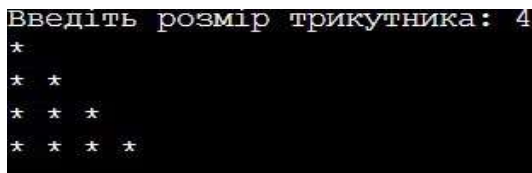
```



```

for (int i = 0; i < n; i++) {
    for (int j = 0; j <= i; j++) {
        printf("* ");
    }
    printf("\n");
}
}

```



```

Введіть розмір трикутника: 4
*
* *
* * *
* * * *

```

4 ДОДАТКОВЕ ЗАВДАННЯ (РІВНОБЕДРЕНИЙ ТРИКУТНИК))

```

#include <stdio.h>

int main() {
    int n;
    printf("Введіть розмір трикутника: ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) {
            printf(" ");
        }
        for (int k = 1; k <= i; k++) {
            printf("* ");
        }
        printf("\n");
    }
}

```

```
Введіть розмір трикутника: 6
  *
 * *
* * *
* * * *
* * * * *
* * * * * *
```

5. Оформити звіт.

Висновок: На даній лабораторній роботі я ознайомився з особливостями операторів циклу в мові програмування C. Використовуючи цикли, ми можемо повторювати певні дії доти, поки виконується певна умова, або просто виконувати дії певну кількість разів. Я розглянув різні види циклів, такі як цикл while, цикл do-while, та цикл for, і навчився використовувати їх у практичних завданнях.