

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОЕЛЕКТРОНІКИ



Лабораторна робота №12

Виконав :

Ст. гр АП-11

Заброварний Олег

Прийняв:

Чайковський І. Б.

Львів 2024

Тема:

Оператори циклу.

Мета:

ознайомитися з особливостями функціонування операторів циклу та навчитись їх використовувати у процесі програмування.

Теоретичні відомості:

У мові C, як і в інших мовах програмування, оператори циклу служать для багаторазового виконання послідовності операторів до тих пір, поки виконується деяка умова. Умова може бути встановлена заздалегідь (як в операторі for) або змінюватися при виконанні тіла циклу (як в while або do-while). Цикл for. У всіх процедурних мовах програмування цикли for дуже схожі. Однак в C цей цикл особливо гнучкий і потужний. Загальна форма оператора for наступна:

Завдання:

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.
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.
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.

5. Оформити звіт.

6. Обчислити скільки зерен необхідно було би видати винахідникові шахів, якщо за першу клітину шахівниці він попросив видати одну зернину пшениці, а за кожну наступну вдвічі більше за попередні. У шахівниці 64 клітини.

7. Для цілих чисел від 1 до 20 обчислити квадратні, кубічні та корені четвертого порядку. Результати звести у таблицю, використовуючи форматування функції printf().

8. Здійснити табулювання функції, що з певними припущеннями з достатньою точністю моделює імпульс Максвелла, який утворюється при ударному збудженні ширококугової антени. Обчислення провести на проміжку зміни i в межах $[0-31]$ з кроком $i=1$, $N=32$. Результати вивести у вигляді таблиці. Визначити найбільше та найменше значення функції на цьому проміжку.

9. В обчислювальних задачах при програмуванні ітераційних алгоритмів, що закінчуються при досягненні заданої точності, часто необхідна оцінка «машинного нуля», тобто числового значення, менше за яке неможливо задати точність даного алгоритму. Абсолютне значення «машинного нуля» залежить від розрядної сітки застосовуваного комп'ютера, від прийнятої в конкретному трансляторі точності представлення дійсних чисел і від значень, що використовуються для оцінки точності. Наступна програма оцінює абсолютне значення «машинного нуля» відносно близьких (за модулем) до одиниці змінних типу float.

10. Обчислити значення скінченної суми, або добутку згідно свого варіанту. Врахувати, що навіть для невеликих чисел значення факторіала може вийти за гранично допустимі для даного типу даних.

В.7 Дано натуральне число N . Обчислити

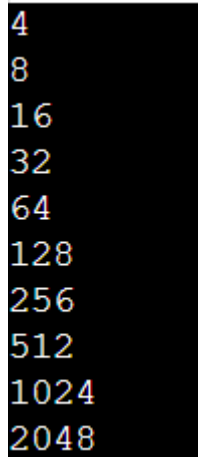
$$S = \sum_{i=1}^N \sum_{k=0}^i (i+k)^2.$$

Виконання роботи:

Завдання 2:

```
#include<stdio.h>

int main() {
    int i = 2;
    while (i <= 1024) {
        i = i * 2;
        printf("%d\n", i);
    }
    return 0;
}
```



A screenshot of a terminal window with a black background and yellow text. It displays the output of the first program, which is a list of powers of 2: 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048, each on a new line.

```
4
8
16
32
64
128
256
512
1024
2048
```

```
#include<stdio.h>

int main() {
    int j = 0, k = 0;
    while (j < 5) {
        k += 10;
        j++;
    }
}
```

```
printf("k=%d\n", k);
```

```
return 0;
```

```
}
```

```
k=50
```

```
#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
```

```
#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);
```

```
return 0;
```

```
}
```

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

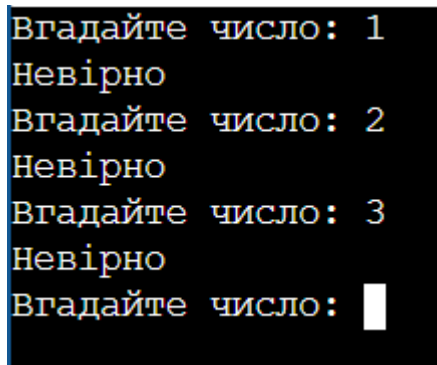
```
#include<stdio.h>
```

```

#include<stdlib.h>

int main(void) {
    int magic;
    int guess;
    magic = rand() % 100 + 1;
    printf("Вгадайте число: ");
    scanf("%d", &guess);
    if(guess == magic) {
        printf("***Вірно! **\n");
        printf("Магічне число рівне %d\n", magic);
    } else {
        printf("Невірно\n");
        while (guess != magic) {
            printf("Вгадайте число: ");
            scanf("%d", &guess);
            printf("Невірно\n");
        }
    }
    return 0;
}

```



```

Вгадайте число: 1
Невірно
Вгадайте число: 2
Невірно
Вгадайте число: 3
Невірно
Вгадайте число: 

```

```

#include<stdio.h>

```

```
#include<stdlib.h>

int main(void) {
    int magic;
    int guess;
    int m = 1;
    magic = rand() % 100 + 1;
    printf("Вгадайте число: ");
    scanf("%d", &guess);
    if(guess == magic) {
        printf("***Вірно! **\n");
        printf("Магічне число рівне %d\n", magic);
    } else {
        printf("Невірно\n");
        printf("m=%d\n", m);
        while (guess != magic) {
            m++;
            printf("Вгадайте число: ");
            scanf("%d", &guess);
            printf("Невірно\n");
            printf("m=%d\n", m);
        }
    }
    return 0;
}
```

```
Вгадайте число: 5
Невірно
m=1
Вгадайте число: 6
Невірно
m=2
Вгадайте число: 7
Невірно
m=3
Вгадайте число: 8
Невірно
m=4
Вгадайте число: 
```

```
#include <stdio.h>

#include <stdlib.h>

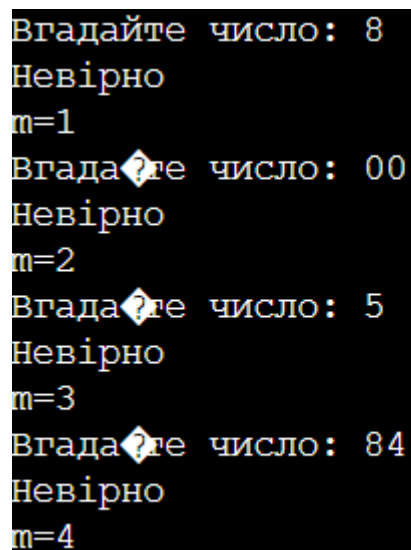
int main(void) {
    int magic;
    int guess;
    int m = 1;
    magic = rand() % 100 + 1;
    printf("Вгадайте число: ");
    scanf("%d", &guess);
    if(guess == magic) {
        printf("***Вірно! **\n");
        printf("Магічне число рівне %d\n", magic);
    } else {
        printf("Невірно\n");
        printf("m=%d\n", m);
        while (guess != magic) {
            m++;
        }
    }
}
```



```

printf("Вгадайте число: ");
scanf("%d", &guess);
printf("Невірно\n");
printf("m=%d\n", m);
if(m > 9) {
printf("Магічне число рівне %d\n", magic);
break;
}
}
}
return 0;
}

```



```

Вгадайте число: 8
Невірно
m=1
Вгадайте число: 00
Невірно
m=2
Вгадайте число: 5
Невірно
m=3
Вгадайте число: 84
Невірно
m=4

```

Завдання 3:

```

#include<stdio.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;
}
return 0;
}

```

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

Завдання 4:

```

#include <stdio.h>

int main() {
    int n, i, j;

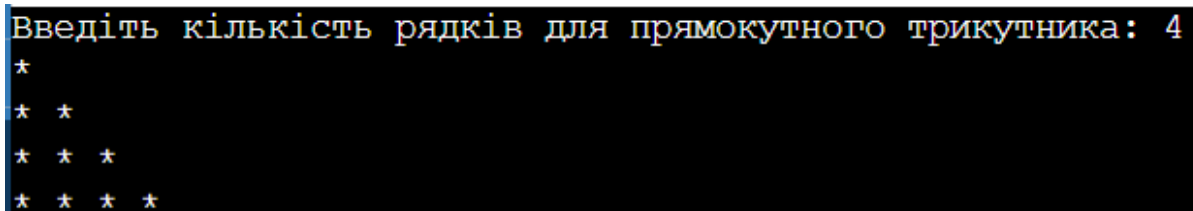
    printf("Введіть кількість рядків для прямокутного трикутника: ");

```

```

scanf("%d", &n);
for (i = 0; i < n; i++) {
    for (j = 0; j <= i; j++) {
        printf("* ");
    }
    printf("\n");
}
return 0;
}

```



```

Введіть кількість рядків для прямокутного трикутника: 4
*
* *
* * *
* * * *

```

Завдання 5:

```

#include <stdio.h>
#include <stdint.h>

int main() {
    int n = 64;
    uint64_t grains = 1;
    uint64_t total_grains = 0;
    for (int i = 0; i < n; i++) {
        total_grains += grains;
        grains *= 2;
    }
    printf("Загальна кількість зерен на шахівниці: %lu\n", total_grains);
}

```

```
    return 0;
}
```

Загальна кількість зерен на шахівниці: 18446744073709551615

Завдання 6:

```
#include <stdio.h>

#include <math.h>

int main() {
    printf("Номер\+квадрат\куб\тчетвертий корінь\n");
    for (int i = 1; i <= 20; i++) {
        double square = pow(i, 2);
        double cube = pow(i, 3);
        double fourth_root = pow(i, 0.25);

        printf("%d\t%.2f\t%.2f\t%.2f\n", i, square, cube, fourth_root);
    }
    return 0;
}
```

Номер	квадрат	куб	четвертий корінь
1	1.00	1.00	1.00
2	4.00	8.00	1.19
3	9.00	27.00	1.32
4	16.00	64.00	1.41
5	25.00	125.00	1.50
6	36.00	216.00	1.57
7	49.00	343.00	1.63
8	64.00	512.00	1.68
9	81.00	729.00	1.73
10	100.00	1000.00	1.78
11	121.00	1331.00	1.82
12	144.00	1728.00	1.86
13	169.00	2197.00	1.90
14	196.00	2744.00	1.93
15	225.00	3375.00	1.97
16	256.00	4096.00	2.00
17	289.00	4913.00	2.03
18	324.00	5832.00	2.06
19	361.00	6859.00	2.09
20	400.00	8000.00	2.11

Завдання 7:

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

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

```
#define PI 3.14159265
```

```
int main() {
```

```
    int N = 32;
```

```
    double max_val = -INFINITY;
```

```
    double min_val = INFINITY;
```

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

```
    for (int i = 0; i <= 31; i++) {
```

```
        double value = i * i * exp(-i * i / 100) * sin((2 * PI / N) * i);
```

```

printf("%d\t%.6f\n", i, value);
if (value > max_val) {
    max_val = value;
}
if (value < min_val) { min_val = value;
    }
}
printf("\nМаксимальне значення: %.6f\n", max_val);
printf("Мінімальне значення: %.6f\n", min_val);
return 0;
}

```

```

18      -6.173070
19      -9.985337
20      -5.180445
21      -6.715944
22      -8.189979
23      -3.495886
24      -3.881057
25      -1.519452
26      -1.548086
27      -0.552729
28      -0.505522
29      -0.156740
30      -0.042504
31      -0.023137

Максимальне значення: 79.443608
Мінімальне значення: -9.985337

```

Завдання 8:

```

#include<stdio.h>

#include<math.h>

int main(void) {

```

```

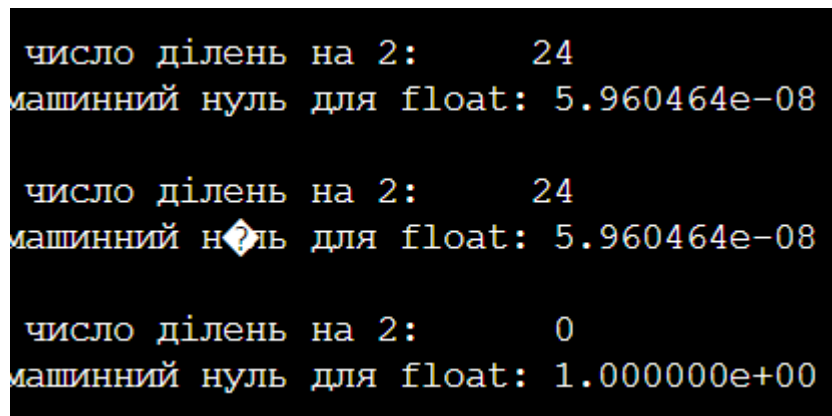
int i = 0;
float precision = 1.0;
float a;
while (1) {
    precision = precision / 2.0;
    a = precision + 1.0;
    i++;
    if (a <= 1.0) {
        break;
    }
}
printf("\n число ділень на 2: %6d\n", i);
printf("машинний нуль для float: %e\n", precision);
i = 0;
precision = 1.0;
do {
    precision = precision / 2.0;
    a = precision + 1.0;
    i++;
} while (a > 1.0);
printf("\n число ділень на 2: %6d\n", i);
printf("машинний нуль для float: %e\n", precision);
i = 0;
precision = 1.0;
for (i = 0; a > 1.0; i++) {
    precision = precision / 2.0;

```

```

    a = precision + 1.0;
}
printf("\n число ділень на 2: %6d\n", i);
printf("машинний нуль для float: %e\n", precision);
return 0;
}

```



```

число ділень на 2:      24
машинний нуль для float: 5.960464e-08

число ділень на 2:      24
машинний нуль для float: 5.960464e-08

число ділень на 2:       0
машинний нуль для float: 1.000000e+00

```

Завдання 9:

```

#include <stdio.h>

int main() {
    int N = 10;
    double S = 0.0;
    for (int i = 1; i < N; i++) {
        double inner_sum = 0.0;
        double factorial = 1.0;
        for(int k = 0; k <= i; k++) {
            double term = factorial / (1.0);
            inner_sum += term;
            factorial *= (i - k);
        }
        S += inner_sum;
    }
}

```



```

    }

    printf("Значення суми S: %.6f\n", S);

    return 0;
}

```

Значення суми S: 1112082.000000

Завдання 10:

```

#include <stdio.h>

#include <math.h>

int main() {

    double x, y_library, y_series, term, a;

    int iterations = 0;

    const double precision = 0.00001;

    printf("Введіть значення x ( $0 \leq x \leq \pi/2$ ): ");

    scanf("%lf", &x);

    y_library = exp(x)-pow(M_E, 2);

    y_series = 0.0;

    a = 1.0;

    term = x;

    while (fabs(term) >= precision) {

        y_series += term;

        iterations++;

        a *= (double)iterations;

        term = pow(x, iterations + 1) / a;

    }

    printf("Значення функції y за допомогою бібліотечної функції: %f\n",
y_library);

```

```
printf("Значення функції у за допомогою ряду: %f\n", y_series);  
printf("Кількість ітерацій: %d\n", iterations);  
return 0;  
}
```

```
Введіть значення x (0 <= x <= pi/2): 5  
Значення функції у за допомогою бібліотечної функції: 141.024103  
Значення функції у за допомогою ряду: 742.065793  
Кількість ітерацій: 23
```

Висновок: На даній лабораторній роботі я ознайомився з особливостями функціонування операторів циклу та навчитись їх використовувати у процесі програмування.