

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

з дисципліни «Програмування»

Підготував:
ст. групи АП-11
Фостик Віталій

Прийняла:
Гордійчук-Бублівська О. В.

Львів – 2024

Тема:

Оператори циклу.

Мета:

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

Теоретичні відомості:

У мові С, 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 С цей цикл особливо гнучкий і потужний. Загальна форма оператора for наступна: for (необов'язковий вираз 1; необов'язковий вираз 2; необов'язковий вираз 3) оператор; Вираз 1 призначений для ініціалізації циклу і виконується один раз. Здебільшого тут задаються початкові значення змінних циклу.

Хід роботи:

1. Ознайомитися з теоретичними відомостями.
2. Здійснити виконання усіх прикладів, представлених у теоретичних відомостях, після чого представити скріншоти 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. Скріншот 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. Скріншот 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 ширококутної антени. Обчислення провести на проміжку зміни і в межах [0-31] з кроком i=1, N=32. Результати вивести у вигляді таблиці.

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.

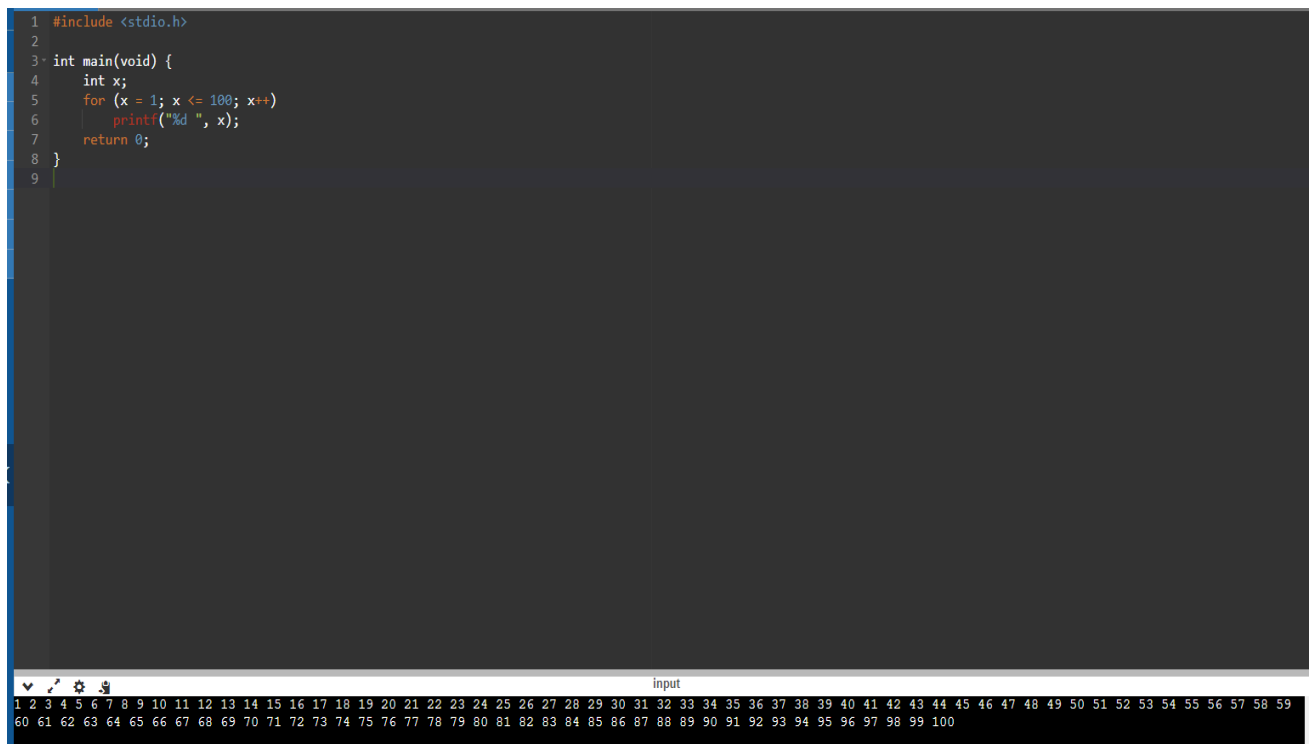
Наступна програма оцінює абсолютне значення «машинного нуля» відносно близьких (за модулем) до одиниці змінних типу float.

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. Аргумент тригонометричних функцій задавати в межах.

Виконання роботи:

2.

```
1 #include <stdio.h>
2
3 int main(void) {
4     int x;
5     for (x = 1; x <= 100; x++)
6         printf("%d ", x);
7     return 0;
8 }
9
```

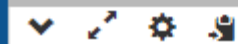


```
1  #include<stdio.h>
2
3  int main() {
4      int i = 2;
5      while (i <= 1024) {
6          i = i * 2;
7          printf("%d\n", i);
8      }
9      return 0;
10 }
11
```



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

```
1  #include<stdio.h>
2
3  int main() {
4      int j = 0, k = 0;
5      while (j < 5) {
6          k += 10;
7          j++;
8      }
9      printf("k=%d\n", k);
10     return 0;
11 }
12
```



k=50

```
1  #include<stdio.h>
2
3  int main() {
4      int a = 2, b = 10, c;
5      do {
6          b = b + a;
7          c = 10 * a;
8          a++;
9      } while (a < 5);
10     printf("\nb=%d c=%d a=%d", b, c, a);
11     return 0;
12 }
13
```


▼ ↗ ⚙ 🐞

b=19 c=40 a=5

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  int main(void) {
4      int magic;
5      int guess;
6      magic = rand() % 100 + 1;
7      printf("Вгадайте число: ");
8      scanf("%d", &guess);
9
10     if(guess == magic) {
11         printf("**Вірно!**\n");
12         printf("Магічне число рівне %d\n", magic);
13     } else {
14         printf("Невірно\n");
15         while (guess != magic) {
16             printf("Вгадайте число: ");
17             scanf("%d", &guess);
18             printf("Невірно\n");
19         }
20     }
21     return 0;
22 }
23

```



Вгадайте число: 1

Невірно

Вгадайте число: 2

Невірно

Вгадайте число: 5

Невірно

Вгадайте число:

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main(void) {
5      int magic;
6      int guess;
7      int m = 1;
8      magic = rand() % 100 + 1;
9      printf("Вгадайте число: ");
10     scanf("%d", &guess);
11
12     if(guess == magic) {
13         printf("***Вірно!**\n");
14         printf("Магічне число рівне %d\n", magic);
15     } else {
16         printf("Невірно\n");
17         printf("m=%d\n", m);
18         while (guess != magic) {
19             m++;
20             printf("Вгадайте число: ");
21             scanf("%d", &guess);
22             printf("Невірно\n");
23             printf("m=%d\n", m);
24         }
25     }
26     return 0;
27 }
28

```

```

Вгадайте число: 5
Невірно
m=18
Вгадайте число: 6
Невірно
m=19
Вгадайте число: 7
Невірно
m=20
Вгадайте число: 8
Невірно
m=21
Вгадайте число:


```



```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void) {
5      int magic;
6      int guess;
7      int m = 1;
8      magic = rand() % 100 + 1;
9      printf("Вгадайте число: ");
10     scanf("%d", &guess);
11     if(guess == magic) {
12         printf("***Вірно!**\n");
13         printf("Магічне число рівне %d\n", magic);
14     } else {
15         printf("Невірно\n");
16         printf("m=%d\n", m);
17         while (guess != magic) {
18             m++;
19             printf("Вгадайте число: ");
20             scanf("%d", &guess);
21             printf("Невірно\n");
22             printf("m=%d\n", m);
23             if(m > 9) {
24                 printf("Магічне число рівне %d\n", magic);
25                 break;
26             }
27         }
28     }
29     return 0;
30 }
31

```



Невірно

m=8

Вгадайте число: 00

Невірно

m=9

Вгадайте число: 5

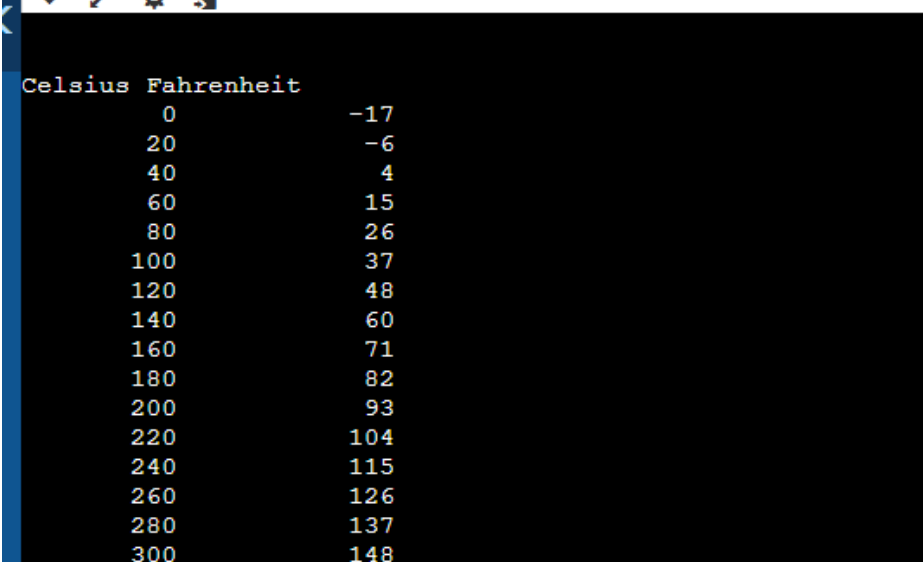
Невірно

m=10

Магічне число рівне 84

3.

```
1  #include<stdio.h>
2
3  int main() {
4      int fahr, celsius;
5      int lower, upper, step;
6
7      lower = 0;
8      upper = 300;
9      step = 20;
10     fahr = lower;
11
12     printf("\n\nCelsius Fahrenheit\n");
13     while (fahr <= upper) {
14         celsius = 5 * (fahr - 32) / 9;
15         printf("%10d\t%8d\n", fahr, celsius);
16         fahr = fahr + step;
17     }
18     return 0;
19 }
20
```



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.

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, j;
5
6     printf("Введіть кількість рядків для прямокутного трикутника: ");
7     scanf("%d", &n);
8
9     for (i = 0; i < n; i++) {
10         for (j = 0; j <= i; j++) {
11             printf("* ");
12         }
13         printf("\n");
14     }
15
16     return 0;
17 }
18
```

input

Введіть кількість рядків для прямокутного трикутника: 4

```
*
* *
* * *
* * * *
```

5.

```
1 #include <stdio.h>
2 #include <stdint.h>
3
4 int main() {
5     int n = 64;
6     uint64_t grains = 1;
7     uint64_t total_grains = 0;
8
9     for (int i = 0; i < n; i++) {
10         total_grains += grains;
11         grains *= 2;
12     }
13
14     printf("Загальна кількість зерен на шахівниці: %lu\n", total_grains);
15
16     return 0;
17 }
18
```

input

Загальна кількість зерен на шахівниці: 18446744073709551615

6.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     printf("Номер\tквадрат\tкуб\tчетвертий корінь\n");
6
7     for (int i = 1; i <= 20; i++) {
8         double square = pow(i, 2);
9         double cube = pow(i, 3);
10        double fourth_root = pow(i, 0.25);
11
12        printf("%d\t%.2f\t%.2f\t%.2f\n", i, square, cube, fourth_root);
13    }
14
15    return 0;
16 }
17
```

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

7.

```

1  #include <stdio.h>
2  #include <math.h>
3  #define PI 3.14159265
4  int main() {
5      int N = 32;
6      double max_val = -INFINITY;
7      double min_val = INFINITY;
8      printf("i\tValue\n");
9
10     for (int i = 0; i <= 31; i++) {
11         double value = i * i * exp(-i * i / 100) * sin((2 * PI / N) * i);
12         printf("%d\t%.6f\n", i, value);
13         if (value > max_val) {
14             max_val = value;
15         }
16         if (value < min_val) {
17             min_val = value;
18         }
19     }
20
21     printf("\nМаксимальне значення: %.6f\n", max_val);
22     printf("Мінімальне значення: %.6f\n", min_val);
23
24     return 0;
25 }

```

```

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

8.

```

1  #include<stdio.h>
2  #include<math.h>
3  int main(void) {
4      int i = 0;
5      float precision = 1.0;
6      float a;
7      while (1) {
8          precision = precision / 2.0;
9          a = precision + 1.0;
10         i++;
11         if (a <= 1.0) {
12             break;
13         }
14     }
15     printf("\n число ділень на 2: %6d\n", i);
16     printf("машинний нуль для float: %e\n", precision);
17     i = 0;
18     precision = 1.0;
19     do {
20         precision = precision / 2.0;
21         a = precision + 1.0;
22         i++;
23     } while (a > 1.0);
24     printf("\n число ділень на 2: %6d\n", i);
25     printf("машинний нуль для float: %e\n", precision);
26     i = 0;
27     precision = 1.0;
28     for (i = 0; a > 1.0; i++) {
29         precision = precision / 2.0;
30         a = precision + 1.0;
31     }
32     printf("\n число ділень на 2: %6d\n", i);
33     printf("машинний нуль для float: %e\n", precision);
34     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

```

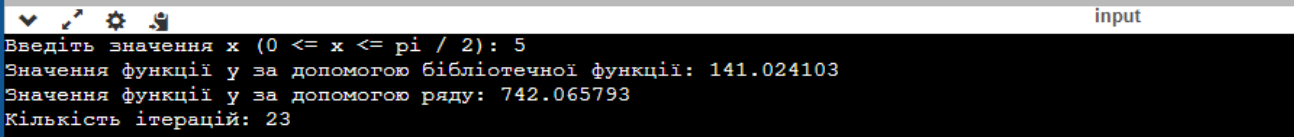
9.

```
1  #include <stdio.h>
2
3  int main() {
4      int N = 10;
5      double S = 0.0;
6      for (int i = 1; i <= N; i++) {
7          double inner_sum = 0.0;
8          double factorial = 1.0;
9          for (int k = 0; k <= i; k++) {
10             double term = factorial / (1.0);
11             inner_sum += term;
12             factorial *= (i - k);
13         }
14         S += inner_sum;
15     }
16     printf("Значення суми S: %.6f\n", S);
17
18     return 0;
19 }
20
```

Значення суми S: 10976183.000000

10.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double x, y_library, y_series, term, a;
6     int iterations = 0;
7     const double precision = 0.00001;
8     printf("Введіть значення x (0 <= x <= pi / 2): ");
9     scanf("%lf", &x);
10    y_library = exp(x) - pow(M_E, 2);
11    y_series = 0.0;
12    a = 1.0;
13    term = x;
14    while (fabs(term) >= precision) {
15        y_series += term;
16        iterations++;
17        a *= (double)iterations;
18        term = pow(x, iterations + 1) / a;
19    }
20    printf("Значення функції y за допомогою бібліотечної функції: %f\n", y_library);
21    printf("Значення функції y за допомогою ряду: %f\n", y_series);
22    printf("Кількість ітерацій: %d\n", iterations);
23
24    return 0;
25 }
26
```



input

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

Контрольні запитання:

Звичайно, допоможу з циклами в програмуванні:

1. Призначення операторів циклу полягає в тому, щоб виконувати певний блок коду або набір інструкцій декілька разів, доки виконуються певні умови.
2. Оператор циклу `while` виконується, доки задана умова є істинною. Спочатку перевіряється умова, і якщо вона істинна, виконується тіло циклу.
3. Оператор циклу `do-while` подібний до `while`, але умова перевіряється після виконання тіла циклу. Таким чином, тіло циклу виконується принаймні один раз, навіть якщо умова вже невірна.

4. Оператор циклу `for` використовується для повторення блоку коду певне число разів. Він складається з трьох основних частин: ініціалізація, умова і крок. Цей цикл часто використовується, коли заздалегідь відомо, скільки разів потрібно виконати цикл.

5. У конструкції циклу `for` вирази мають такі призначення:

- Ініціалізаційний вираз: виконується один раз на початку циклу і встановлює початкові значення змінних.

- Умова: перевіряється перед кожним виконанням тіла циклу і вказує, чи має цикл продовжуватися.

- Крок: виконується після кожного виконання тіла циклу і використовується для зміни значень змінних керування циклом.