

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут телекомунікацій, радіоелектроніки та електронної техніки



ЗВІТ
з лабораторного практикуму
з дисципліни «Програмування»

Підготував
Студент групи АП-11
Ніжник Станіслав
Прийняла
Гордійчук – Бублівська О. В.

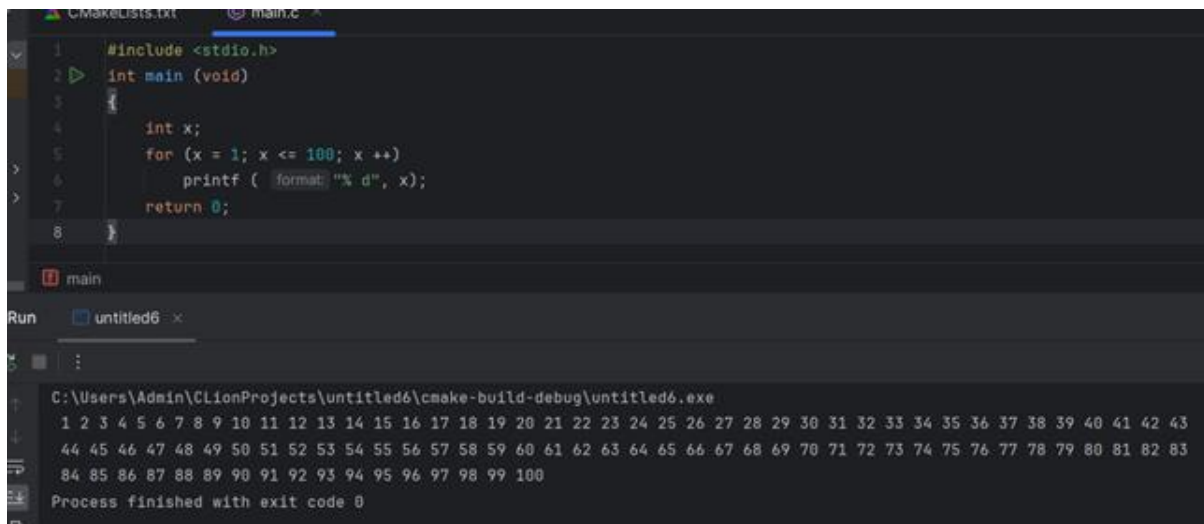
Львів – 2024

Лабораторна робота №12

Тема: Оператори циклу

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

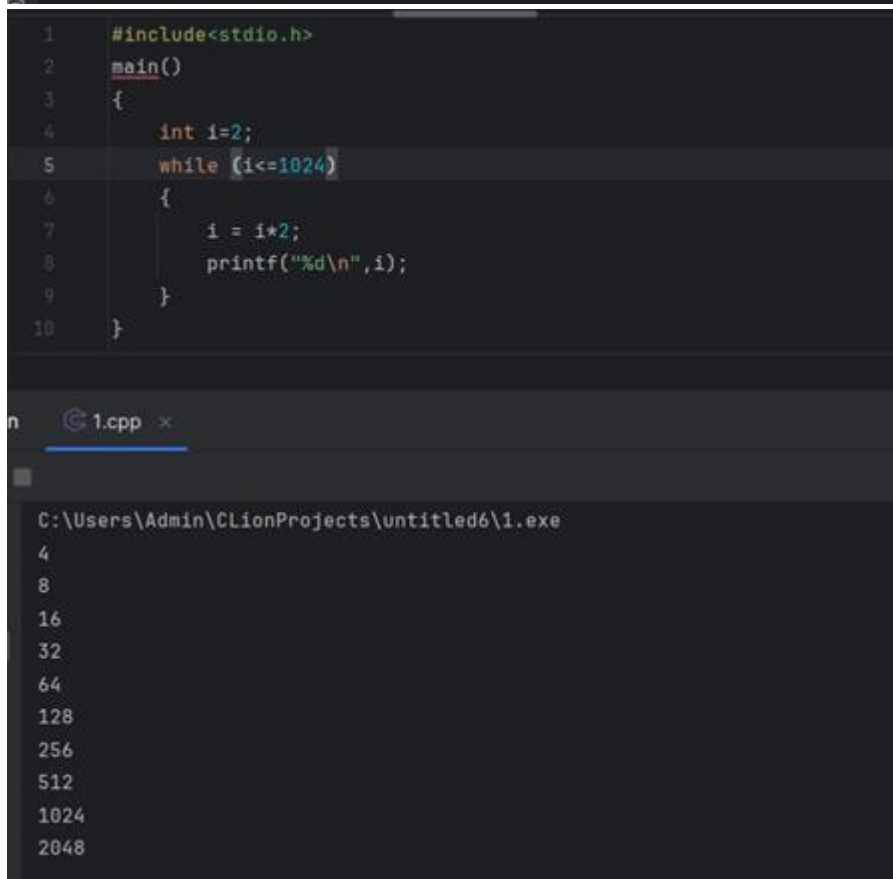
1. Здійснити виконання усіх прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати їх виконання у звіті.



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

Run untitled6 x

```
C:\Users\Admin\CLionProjects\untitled6\cmake-build-debug\untitled6.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
Process finished with exit code 0
```



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

n 1.cpp x

```
C:\Users\Admin\CLionProjects\untitled6\1.exe
4
8
16
32
64
128
256
512
1024
2048
```



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

main

2.cpp x

C:\Users\Admin\CLionProjects\untitled6\2.exe

k=50

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  int main(void)
4  {
5      int magic;
6      int guess;
7      magic=rand();
8      printf( format: "vgaday chuslo:");
9      scanf( format: "%d",&guess);
10     if(guess==magic){
11         printf( format: "***virno**");
12         printf( format: "magichne chuslo rive %d",magic);
13     }
14     else {
15         printf( format: "nevirno\n");
16         while (guess!=magic){
17             printf( format: "vgaday chuslo:");
18             scanf( format: "%d",&guess);
19             printf( format: "nevirno\n");
20         }
21     }
22 }
```

main

Run 4.cpp x

C:\Users\Admin\CLionProjects\untitled6\4.exe

vgaday chuslo:6

nevirno

vgaday chuslo:|

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  int main(void)
4  {
5      int magic;
6      int guess;
7      int m=1;
8      magic=rand();
9      printf( format: "vgaday chuslo:");
10     scanf( format: "%d",&guess);
11     if(guess==magic){
12         printf( format: "**virno**");
13         printf( format: "magichne chuslo rivne %d",magic);
14     }
15     else {
16         printf( format: "nevirno\n");
17         printf( format: "m=%d\n",m);
18         while (guess!=magic){
19             m++;
20             printf( format: "vgaday chuslo:");
21             scanf( format: "%d",&guess);
22             printf( format: "nevirno\n");
23             printf( format: "m=%d\n",m);
24         }
25     }
26 }

```

```

C:\Users\Admin\CLionProjects\untitled6\5.exe
vgaday chuslo:40
nevirno
m=1
vgaday chuslo:|

```

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  int main(void)
4  {
5      int magic;
6      int guess;
7      int m=1;
8      magic=rand();
9      printf( format: "vgaday chuslo:");
10     scanf( format: "%d",&guess);
11     if(guess==magic){
12         printf( format: "**virno**");
13         printf( format: "magichne chuslo rivne %d",magic);
14     }
15     else {
16         printf( format: "nevirno\n");
17         printf( format: "m=%d\n",m);
18         while (guess!=magic){
19             m++;
20             printf( format: "vgaday chuslo:");
21             scanf( format: "%d",&guess);
22             printf( format: "nevirno\n");
23             printf( format: "m=%d\n",m);
24         }
25     }
26 }
27 |

```

```

C:\Users\Admin\CLionProjects\untitled6\5.exe
vgaday chuslo:1
nevirno
m=1
vgaday chuslo:|

```

1. 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.

```

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

3. 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.

```

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

```

main

15.cpp x 5.cpp x

C:\Users\Admin\CLionProjects\untitled6\15.exe

Введіть значення n: 6

```

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

```

4. 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

```
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      int num_cells = 64;
5      unsigned long long int total_grains;
6      total_grains = (unsigned long long int) pow(2, num_cells) - 1;
7      printf("Загальна кількість зерен пшениці для винахідника шахів: %llu\n", total_grains);
8
9      return 0;
10 }
11
```

C:\Users\Admin\CLionProjects\untitled6\14.exe
Загальна кількість зерен пшениці для винахідника шахів: 18446744073709551615

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

```
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      printf("Число\tКвадрат\tКуб\tКорінь 4\n");
5      printf("-----\n");
6      for (int i = 1; i <= 20; i++) {
7          int square = i * i;
8          int cube = i * i * i;
9          double root_4 = pow(i, 0.25);
10         printf("%d\t%d\t%d\t%.2f\n", i, square, cube, root_4);
11     }
12 }
```

C:\Users\Admin\CLionProjects\untitled6\16.exe

Число	Квадрат	Куб	Корінь 4

1	1	1	1.00
2	4	8	1.19
3	9	27	1.32
4	16	64	1.41
5	25	125	1.50
6	36	216	1.57
7	49	343	1.63
8	64	512	1.68
9	81	729	1.73
10	100	1000	1.78
11	121	1331	1.82
12	144	1728	1.86
13	169	2197	1.90
14	196	2744	1.93
15	225	3375	1.97
16	256	4096	2.00
17	289	4913	2.03
18	324	5832	2.06
19	361	6859	2.09
20	400	8000	2.11

6. 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 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

```

1  #include <stdio.h>
2  #include <math.h>
3  double func(int i) {
4      return pow(1, y/2) * exp(-1 / 100) * sin(X/2 * M_PI * i);
5  }
6  int main() {
7      int start = 0;
8      int end = 31;
9      int step = 1;
10     int N = (end - start + 1) / step;
11     printf("i\ty\n");
12     for (int i = start; i <= end; i += step) {
13         double y = func(i);
14         printf("i\ty\n", i, y);
15     }
16     double max = -INFINITY;
17     double min = INFINITY;
18     for (int i = start; i <= end; i += step) {
19         double y = func(i);
20         if (y > max) {
21             max = y;
22         }
23         if (y < min) {
24             min = y;
25         }
26     }
27     printf("Максимальне значення: %.5f\n", max);
28     printf("Мінімальне значення: %.5f\n", min);
29 }

```

C:\Users\Admin\CLionProjects\untitled6\8.exe

i	y
0	0.00000
1	-0.00000
2	-0.00000
3	-0.00000
4	-0.00000
5	-0.00000
6	-0.00000
7	-0.00000
8	-0.00000
9	-0.00000
10	-0.00000

11	-0.00000
12	-0.00000
13	0.00000
14	-0.00000
15	-0.00000
16	-0.00000
17	0.00000
18	-0.00000
19	-0.00000
20	-0.00000
21	0.00000
22	-0.00000
23	-0.00000
24	-0.00000
25	0.00000
26	0.00000
27	-0.00000
28	-0.00000
29	0.00000
30	-0.00000
31	-0.00000

Максимальне значення: 0.00000

Мінімальне значення: -0.00000

Process finished with exit code 0

7. 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. Завдання: змінити програму застосовувавши кожного разу один із трьох циклічних операторів. Оцінку «машинного нуля» провести також для даних типу double -формат виведення %le, longdouble формат виведення %Le.

```
1 #include <math.h>
2 int main(void) {
3     int i;
4     float precision_float, a_float;
5     printf("Для типу float:\n");
6     i = 0;
7     precision_float = 1.0;
8     while ((precision_float / 2.0) + 1.0 > 1.0) {precision_float /= 2.0;
9         i++;
10    }
11    printf("Число ділень на 2: %d\n", i);
12    printf("Машинний нуль: %e\n", precision_float);
13    printf("Для типу double:\n");
14    i = 0;
15    double precision_double, a_double; precision_double = 1.0;
16    while ((precision_double / 2.0) + 1.0 > 1.0) {precision_double /= 2.0;
17        i++;
18    }
19    printf("Число ділень на 2: %d\n", i);
20    printf("Машинний нуль: %e\n", precision_double);
21    printf("Для типу long double:\n");
22    i = 0;
23    long double precision_long_double, a_long_double; precision_long_double = 1.0;
24    while ((precision_long_double / 2.0) + 1.0 > 1.0) {precision_long_double /= 2.0;
25        i++;
26    }
27    printf("Число ділень на 2: %d\n", i);
28    printf("Машинний нуль: %le\n", precision_long_double);
29 }
30 }
```

```
C:\Users\Admin\CLionProjects\untitled6\9.exe
```

```
Для типу float:
```

```
Число ділень на 2: 52
```

```
Машинний нуль: 2.220446e-16
```

```
Для типу double:
```

```
Число ділень на 2: 52
```

```
Машинний нуль: 2.220446e-16
```

```
Для типу long double:
```

```
Число ділень на 2: 63
```

```
Машинний нуль: 1.084202e-19
```

```
Process finished with exit code 0
```

8. 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: $0 \leq X \leq \pi / 2$.

В.10 Дано натуральне число N і дійсне x . Обчислити

$$S_1 = \sum_{i=1}^N (\sin x)^i; S_2 = \sum_{i=1}^N \sin x^i.$$

```
1  #include <stdio.h>
2  #include <math.h>
3  double sum_sin_powers(int n, double x) {
4      double total_sum = 0;
5      for (int i = 1; i <= n; i++) {
6          total_sum += pow(x * sin(x), i);
7      }
8      return total_sum;
9  }
10 int main() {
11     int n = 10;
12     double x = M_PI / 4;
13     double s1 = sum_sin_powers(n, x);
14     printf("Сума S1: %f\n", s1);
15 }
16
```

main

un 10.cpp x 5.cpp x

C:\Users\Admin\CLionProjects\untitled6\10.exe
Сума S1: 2.338769

Process finished with exit code 0

Висновок: в даній лабораторній роботі я ознайомилась з особливостями функціонування операторів циклу та навчилась їх використовувати у процесі програмування.