Міністерство освіти і науки України

Інститут телекомунікацій, радіоелектроніки та електронної техніки

Звіт до лабораторної роботи № 12

Тема:« Оператори циклу»

з дисципліни «Програмування частина 2»

Варіант № 6

виконав студент групи АП-11

Головацький Назар

перевірив доцент кафедри ТК

Чайковський І.Б

Львів 2024

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

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.

```
#include<stdio.h>
#include<conio.h>
main()
{
int fahr,celsius;
int lower,upper,step;
lower=0;
upper=300;
step=20;
fahr=lower;
printf("\n\nCelsius Fahrengeit\n");
while( fahr <= upper )
{ celsius = 5*(fahr-32)/9;
printf("%10d\t%8d\n",fahr,celsius);
fahr=fahr+step;
}
getch();
}
```

Результат:

Celsius Fahrengeit

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 висоту трикутника: ");
    scanf("%d", &n);

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

    return 0;
}
```

Результат:

```
Введіть висоту трикутника: 7
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
```

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

```
#include <stdio.h>

#include <math.h>


int main() {

    int n = 64;

    double grains = pow(2, n) - 1;

    printf("Сумарна кількість зерен на шахівниці: %.0lf\n", grains);


    return 0;

}
```

Результат:

Сумарна кількість зерен на шахівниці: 18446744073709551616

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

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

int main() {
    printf("  Число  |  Квадрат  |  Куб  |  Корінь 4\n");
    printf("-----------------------------------------\n");
    for(int i = 1; i <= 20; i++) {
        double square = pow(i, 2);
        double cube = pow(i, 3);
        double fourth_root = pow(i, 1.0 / 4);
        printf("    %2d    |  %8.0lf  | %6.0lf | %10.2lf\n", i, square, cube, fourth_root);
    }
    return 0;
}
```

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

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

#define N 32

double maxwell_impulse(int i) {
    return pow(i, 2) * exp(-pow(i, 2) / 100) * sin(2 * M_PI * i / N);
```

```
}

int main() {
    printf("  i  |   y(i)\n");
    printf("--------------\n");

    double max_val = -INFINITY;
    double min_val = INFINITY;

    for (int i = 0; i <= 31; i++) {
        double y = maxwell_impulse(i);
        printf("%3d  |  %8.2lf\n", i, y);



        if (y > max_val) {
            max_val = y;
        }
        if (y < min_val) {
            min_val = y;
        }
    }
    printf("\nНайбільше значення: %.2lf\n", max_val);
    printf("Найменше значення: %.2lf\n", min_val);

    return 0;
}
```

Результат:

i	y(i)
0	0.00
1	0.19
2	1.47
3	4.57
4	9.64
5	16.19
6	23.20
7	29.44
8	33.75
9	35.34
10	33.99
11	30.00
12	24.12
13	17.32
14	10.57
15	4.63
16	0.00
17	-3.13
18	-4.86
19	-5.43
20	-5.18
21	-4.46
22	-3.54
23	-2.62

24 | -1.82

25 | -1.18

26 | -0.72

27 | -0.41

28 | -0.22

29 | -0.10

30 | -0.04

31 | -0.01

Найбільше значення: 35.34

Найменше значення: -5.43

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 (КОД)

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

#include <stdio.h>

```
#include <math.h>

int main() {
    int i;
    float precision_float = 1.0f;
    double precision_double = 1.0;
    long double precision_long_double = 1.0L;

    // Для float
    printf("Для змінної типу float:\n");
    for (i = 0; precision_float + 1.0f > 1.0f; i++) {
        precision_float /= 2.0f;
    }
    printf("Число ділень на 2: %d\n", i);
    printf("Машинний нуль: %e\n", precision_float);

    // Для double
    printf("\nДля змінної типу double:\n");
    i = 0;
    while (precision_double + 1.0 > 1.0) {
        precision_double /= 2.0;
        i++;
    }
    printf("Число ділень на 2: %d\n", i);
    printf("Машинний нуль: %le\n", precision_double);
```

```
    // Для long double
    printf("\nДля змінної типу long double:\n");
    i = 0;
    do {
        precision_long_double /= 2.0L;
        i++;
    } while (precision_long_double + 1.0L > 1.0L);
    printf("Число ділень на 2: %d\n", i);
    printf("Машинний нуль: %Le\n", precision_long_double);

    return 0;
}
```

Результат:

Для змінної типу float:

Число ділень на 2: 24

Машинний нуль: 5.960464e-08

Для змінної типу double:

Число ділень на 2: 53

Машинний нуль: 1.110223e-16

Для змінної типу long double:

Число ділень на 2: 64

Машинний нуль: 5.421011e-20

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:

```
#include <stdio.h>

int main() {
    int N, i, k;
    long long S = 0;

    N = 10;

    for(i = 1; i <= N; i++) {
        for(k = 0; k <= i; k++) {
            S += (i + k) * (i + k);
        }
    }

    printf("The sum S is: %lld\n", S);

    return 0;
}
```

Результат:

```
The sum S is: 8030
```

11. Відомо, що одним із методів обчислення багатьох функцій є розкладання їх в ряд Тейлора:

Завдання: для заданого x, яке уводиться з клавіатури під час роботи програми, обчислити значення функції уза допомогою бібліотечних функцій компілятора (наприклад e x – pow(e,2)), а такожза допомогою вище наведеного явного розкладу її в ряд (ітераційний процес до досягнення заданої точності). Обчислити при цьому також кількість ітерацій, тобто кількість членів ряду в розкладі функції. Точність обчислень, тобто значення члена ряду розкладу функції коли необхідно припиняти ітераційний процес, яке в кожній ітерації обчислюється як різниця між бібліотечним значенням і членом розкладу, a=0.00001. Аргумент тригонометричних функцій задавати в межах: / 2.π≤ X ≤0

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

int factorial(int n) {
    int fact = 1;
    for(int i = 1; i <= n; i++) {
        fact *= i;
    }
    return fact;
}

double calculate_ex(double x, double a) {
    double sum = 1.0;  // Initialize sum of series
    double term = 1.0;
    int i = 1;
    while(fabs(term) > a) {
        term *= x / i;
        sum += term;
```

```
        i++;
    }
    return sum;
}

int main() {
    double x, a = 0.00001;
    printf("Enter the value of x (0 <= x <= pi/2): ");
    scanf("%lf", &x);



    double library_value = exp(x);
    printf("The value of e^x calculated using the library function is %.5lf.\n",
library_value);


    double taylor_value = calculate_ex(x, a);
    printf("The value of e^x calculated using the Taylor series is %.5lf.\n",
taylor_value);


    return 0;
}
```