

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №5
" ФУНКЦІЇ"
Варіант №9

Розробив студент гр. 1 БС-22

Сосницький В.О.

Лабораторну роботу захищено

з оцінкою _____

Перевірив

ст. викл. каф.ЗІ_____ Кирилащук Т.Г.

_____ 2022 р.

Вінниця ВНТУ 2022р

Завдання 1

9	<p>Напишіть програму, яка допоможе школяру вивчити таблицю множення. Згенеруйте два додатних однорозрядних числа (окрема функція). Програма виводить, наприклад, питання: "Скільки буде 5 на 6?".</p> <p>Школяр повинен відповісти. Якщо відповідь правильна, програма друкує</p> <p><i>"Молодець! Дуже добре!"</i></p> <p>і після цього задає наступне питання на множення. Якщо відповідь неправильна, програма друкує: "Невірно! Спробуйте знову ...", до тих пір, поки відповідь не буде правильною. Розроблювана функція повинна отримувати три числа: два множники та число-відповідь школяра, і друкувати потрібну фразу.</p>
---	--

Формалізація:

Вхідні дані: answer_user, repeat.

Вихідні дані: number[0], number[1]

Лістинг програми

```
#include <iostream>
#include <ctime>

using namespace std;

void NumberGeneration(int number[2])
{
    srand(time(0));

    for (int i = 0; i < 2; i++) {
        number[i] = 1 + rand() % 9;
    }
}

void MainFunctionCalculation(int number_1, int number_2, int answer)
{
    int summ = number_1 * number_2;
    do
    {
        if (answer == summ) {
            cout << "Молодец! Очень хорошо! " << endl;
        }
        else {
            cout << "Не верно , повтори еще раз! " << endl;
            cin >> answer;
        }
    } while (answer != summ);
}

int main()
{
    int number[2];
    int answer_user;
    int repeat = 0;
    while (repeat != -1)
    {
        NumberGeneration(number);
        cout << "Сколько будет " << number[0] << " умноженое на " << number[1] << "
?" << "\n" << "Для завершения введите -1" << endl;
        cin >> answer_user;
        MainFunctionCalculation(number[0], number[1], answer_user);
        cout << "Хотите еще раз сыграть? Если да , то введите 1, если нет то -1" <<
endl;
        cin >> repeat;
    }
    system("pause");
}
```

```
    return 0;
}
```

Контрольний приклад

```
Сколько будет 5 умноженое на 1 ?
Для завершение введите -1
5
Молодец! Очень хорошо!
Хотите еще раз сыграть? Если да , то введите 1,  если нет то -1
1
Сколько будет 1 умноженое на 2 ?
Для завершение введите -1
2
Молодец! Очень хорошо!
Хотите еще раз сыграть? Если да , то введите 1,  если нет то -1
1
Сколько будет 9 умноженое на 6 ?
Для завершение введите -1
1
Не верно , повтори еще раз!
```

```
Сколько будет 4 умноженое на 1 ?
Для завершение введите -1
4
Молодец! Очень хорошо!
Хотите еще раз сыграть? Если да , то введите 1,  если нет то -1
1
Сколько будет 2 умноженое на 6 ?
Для завершение введите -1
12
Молодец! Очень хорошо!
Хотите еще раз сыграть? Если да , то введите 1,  если нет то -1
-1
Для продолжения нажмите любую клавишу . . . ■
```

Завдання2

Задача 2. Розробити програму, яка працюватиме наступним чином.

- 1) Головна функція програми повинна викликати виконання окремих завдань з попередніх лабораторних робіт (для цього підготувати окрему функцію-меню, яка повертає номер вибраного завдання для виконання). Доцільно для вибору завдання використати оператор вибору.
- 2) Програма повинна працювати нескінченно, поки користувач не захоче припинити її виконання (передбачити відповідний пункт меню).
- 3) Передбачити розробку таких функцій:
 - функція, яка за вказаним кутом обчислює вказаний вираз – лабораторна робота 1 (задача А);
 - функції для обчислення $Y1$ і $Y2$ – для виконання завдання лабораторної роботи №2 (задача 1);
 - функції, які повертають випадкове число (ціле і дійсне – дві окремі функції) з вказаного діапазону (границі інтервалу – аргументи функції) – лабораторна робота №3 (задача 1);
 - функція для генерування одновимірного масиву і функція виведення його на екран – для виконання завдання лабораторної роботи №3 (задача 3, але масив повинен бути динамічним);
 - функція для генерування двовимірного масиву і функція для виведення двовимірного масиву для виконання завдання лабораторної роботи №4 (задача 1).

Формалізація:

Вхідні дані: a, x, k, m, n, b, sizeN4, sizeM4

Вихідні дані: Lab1(a) Lab2(x, a) Integers(k, m);Floats(k, n);

generate_mas3(size3, a, b, Arr);

print_mas3(size3, Arr);

generate_mas4(sizeN4, sizeM4, ARR, a, b);

printmas4(sizeN4, sizeM4, ARR);

Лістинг програми

```
#include <iomanip>
#include <iostream>
#include <windows.h>
```

```
using namespace std;
```

```
int Bar() {
```

```
    int n = 0, num=0;
    cout << "\t 1. Обчислення тригонометричної функції.\n\t 2. Обчислення виразу.\n\t 3. Випадкові цілі й дійсні числа.\n\t 4.У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.\n\t 5.Сума елементів рядків двовимірного масиву\n\t ";
```

```

    cout << "\t Для того, щоб завершити напиши 0.";
    cout << "\n Задача: ";
    cin >> num;
    while (n > 5) {
        cout << " \033[91mМожна виконати лише 5 задач!\n";
        cout << " Яку задачу виконати?";
        cout << "\n Задача: ";
        cin >> num;
    }
    while (n < 0) {
        cout << " \033[91mЦе виконати неможливо!\n";
        cout << " Яку задачу виконати?";
        cout << "\n Задача: ";
        cin >> n;
    }
    cout << endl;
    return num;
}

double Lab1(double a)
{
    const float PI = 3.14;
    float z1, z2, z;
    double sin = 0;
    a = PI / 180;
    z1 = pow(sin * PI / 2 - 3 * a, 2);
    z2 = 1 - sin * (3 * a - PI);
    z = z1 / z2;
    return z;
}

int Lab2( int x, int a)
{
    int y, y1, y2;
    if (x % 3 == 2)
        y1 = a * x;

    else { y1 = 9; }

    if (a > x && a <= x)
        y2 = a - x;
    else
        y2 = a + 2;
    y = y1 * y2;
    return y;
}

void Integers(int k, int m) {
    int a = -33, b = 333;
    cout << " Випадкові цілі числа: " << endl;
    for (int i = 1; i <= m; i++) {
        int x = rand() % (b - a) + a;
        cout << setw(5) << x;
        if (i % k == 0)
            cout << "\n";
    }
    cout << endl;
}

void Floats(int n, int k) {
    int A = -1, B = 2;
    cout << "\nСкільки дійсних чисел згенерувати ?" << endl;
    cin >> n;
    for (int i = 0; i < n; i++)

```

```

        {
            float y = (double)rand() / RAND_MAX * ((B - A) + A);
            if (i % k == 0)
                cout << endl;
        }
    }

void generate_mas3(int n, int a, int b, int* Arr) {
    for (int i = 0; i <= n; i++) {
        Arr[i] = rand() % (b - a) + a;
        cout << Arr[i]<< setw(5)<< "\t" << endl;
    }
    cout << endl;
}

void print_mas3(int size3, int* Arr) {
    cout << " Массив: \n";
    int k = 1;
    for (int i = 0; i <= size3; i++) {
        if (Arr[i] < 0) {
            Arr[i] *= -1;
            cout << setw(5) << Arr[i];
        }
        else {
            Arr[i] = pow(Arr[i], 3);
            cout << setw(5) << Arr[i];
        }
        if (k % 10 == 0)
        {
            cout << endl;
        }
        k++;
    }
}

const int N4 = 20, M4 = 20;
void generate_mas4(int sizeN4, int sizeM4, int ARR[N4][M4], int a, int b) {
    for (int i = 0; i < sizeN4; i++) {
        for (int j = 0; j < sizeM4; j++) {
            ARR[i][j] = rand() % (b - a) + a;
        }
    }
}

void printmas4(int sizeN4, int sizeM4, int ARR[N4][M4]) {
    for (int i = 0; i < sizeN4; i++) {
        for (int j = 0; j < sizeM4; j++) {
            cout << setw(5) << ARR[i][j];
            if (j == sizeN4 - 1)
                cout << endl;
        }
    }
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    do {
        int num = Bar();
        if (num == 0)
            return 0;
        switch (num) {
            case 1: {
                double a;
                cout << " 1. Обчислення тригонометричної функції." << "\n";
                cout << " Введіть кут A: ";
                cin >> a;
                cout << " Z = " << Lab1(a) << endl;
            }
        }
    } while (true);
}

```

```

        break;
    }
    case 2: {
        int x, y, a;
        cout << " 2. Обчислення виразу." << "\n";
        cout << " Введіть x: ";
        cin >> x;
        cout << " Введіть a: ";
        cin >> a;
        cout << " y = " << Lab2(x, a) << endl;
        break;
    }
    case 3: {
        int m, n, k;
        cout << " 3. Випадкові цілі й дійсні числа." << "\n";
        cout << " Кількість чисел в рядку: ";
        cin >> k;
        while (k < 3 || k > 10) {
            cout << " \033[91mЧисло має бути від 3 до 10!\n";
            cout << "\n Кількість чисел в рядку: ";
            cin >> k;
        }
        cout << "\n КІЛЬКІСТЬ ЦІЛИХ ЧИСЕЛ: ";
        cin >> m;
        Integers(k, m);
        cout << "\n КІЛЬКІСТЬ ДІЙСНИХ ЧИСЕЛ: ";
        cin >> n;
        Floats(k, n);
        break;
    }
    case 4: {
        int a, b;
        const int size3 = 10;
        cout << " 4. У всіх від'ємних елементів поміняти знак, а з
додатніх добути корінь кубічний" << "\n";
        cout << " Введіть інтервал елементів масиву через пробіл: ";
        cin >> a >> b;
        while (a >= b) {
            cout << " \033[91mПерша межа має бути менша за другу!!";
            cout << "\n Введіть інтервал елементів масиву через
пробіл: ";
            cin >> a >> b;
        }
        int* Arr = new int[size3];
        generate_mas3(size3, a, b, Arr);
        print_mas3(size3, Arr);
        break;
    }
    case 5: {
        const int N4 = 20, M4 = 20;
        int sumArr[M4], a, b;
        int ARR[N4][M4];
        int sizeN4, sizeM4;
        cout << " 5. Сума елементів рядків двовимірного масиву." <<
"\n";

        cout << " Введіть розмірність масиву через пробіл [n] [m]: ";
        cin >> sizeN4 >> sizeM4;
        while (sizeN4 >= N4 || sizeM4 >= M4 || sizeN4 < 0 || sizeM4 < 0)
        {
            cout << " \033[91mРозмірність має бути не менше 0 та не
більше 20!\n";
            cout << "\n Введіть розмірність масиву через пробіл [N]
[M]: ";
            cin >> sizeN4 >> sizeM4;
        }
    }

```

```

cout << " Введіть інтервал елементів масиву через пробіл: ";
cin >> a >> b;
while (a >= b) {
    printf(" \033[91mПерша межа має бути менша за другу!!");
    cout << "\n Введіть інтервал елементів масиву через

    пробіл: ";

    cin >> a >> b;
}

cout << "\n Масив: \n";
generate_mas4(sizeN4, sizeM4, ARR, a, b);
printmas4(sizeN4, sizeM4, ARR);
cout << "\n Результат: \n";
for (int i = 0; i < sizeN4; i++) {
    sumArr[i] = 0;
    for (int j = 0; j < sizeM4; j++) {
        sumArr[i] += ARR[i][j];
        cout << setw(4) << sumArr[i];

    }
    cout << endl;

}

}
}
} while (true);
}

```

Контрольний приклад

1. Обчислення тригонометричної функції.
 2. Обчислення виразу.
 3. Випадкові цілі й дійсні числа.
 4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.
 5. Сума елементів рядків двовимірного масиву
 Для того, щоб завершити написи 0.

Задача: 1

1. Обчислення тригонометричної функції.

Введіть кут A: 30

Z = 0.06273678

1. Обчислення тригонометричної функції.

2. Обчислення виразу.

3. Випадкові цілі й дійсні числа.

4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.

5. Сума елементів рядків двовимірного масиву

Для того, щоб завершити написи 0.

Задача: 2

2. Обчислення виразу.

Введіть x: 2

Введіть a: 3

y = 30

1. Обчислення тригонометричної функції.

2. Обчислення виразу.

3. Випадкові цілі й дійсні числа.

4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.

5. Сума елементів рядків двовимірного масиву

Для того, щоб завершити написи 0.

Задача: 3

Задача: 4

4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний
 Введіть інтервал елементів масиву через пробіл: -50 50

21

-12

19

-38

17

49

-15

14

-47

-39

-28

```

5. Сума елементів рядків двовимірного масиву.
Введіть розмірність масиву через пробіл [n] [m]: 10
20
Розмірність має бути не менше 0 та не більше 20!

Введіть розмірність масиву через пробіл [N] [M]: 10
15
Введіть інтервал елементів масиву через пробіл: -50 50

Масив:
-17  23  14  -9 -39   3  18  -3  -6  12
  7 -13   9 -27  -9 -21  28 -34 -15  40  -8  38 -44 -10  -8
 14  -2  -4 -45  40 -21  20   0 -44 -49  43  -2 -21 -27  34
  4   6 -10  16  26 -19 -42  -6 -11 -24 -27 -13 -12 -32  32
-21  -9 -17 -35 -11   8 -46 -20  27 -44  23  36 -29  -5 -26
 22  20 -21  27  23  47 -38  36  40  11 -14   5  17   5  24
-19   2   0   0  -9 -26  16 -20 -43  41 -43 -13   7  37   3
 33  -5 -41 -41   8 -29  38 -28  -4 -44 -20 -37  18 -50  41
 12   5 -40   9 -26 -13  -2  33  45  -9 -48   0  41 -14  24
-30  46 -29  -2  49  18  34  31 -16   3  49 -32 -12 -50  38
-23  17 -22  43  -2

Результат:
-17   6  20  11 -28 -25  -7 -10 -16  -4   3 -10  -1 -28 -37
-21   7 -27 -42  -2 -10  28 -16 -26 -34 -20 -22 -26 -71 -31
-21  -1  -1 -45 -94 -51 -53 -74-101 -67 -63 -57 -67 -51 -25
-19 -61 -67 -78-102-129-142-154-186-154-175-184-201-236-247
   8 -38 -58 -31 -75 -52 -16 -45 -50 -76 -54 -34 -55 -28  -5
  47   9  45  85  96  82  87 104 109 133 114 116 116 116 107
-26 -10 -30 -73 -32 -75 -88 -81 -44 -41  -8 -13 -54 -95 -87
-29   9 -19 -23 -67 -87-124-106-156-115-103 -98-138-129-155
-13 -15  18  63  54   6   6  47  33  57  27  73  44  42  91
 18  52  83  67  70 119  87  75  25  63  40  57  35  78  76

1. Обчислення тригонометричної функції.
2. Обчислення виразу.
3. Випадкові цілі й дійсні числа.
4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.
5. Сума елементів рядків двовимірного масиву
   Для того, щоб завершити напиши 0.

Задача:

```

```

1. Обчислення тригонометричної функції.
2. Обчислення виразу.
3. Випадкові цілі й дійсні числа.
4. У всіх від'ємних елементів поміняти знак, а з додатніх добути корінь кубічний.
5. Сума елементів рядків двовимірного масиву
   Для того, щоб завершити напиши 0.

Задача: 0

C:\Users\user\source\repos\5lab 2 zadanie\x64\Debug\5lab 2 zadanie.exe (процесс 1032) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

Висновок:

Досліджено принципи побудови функцій користувача, основні складові функцій, прототипи, оголошення та опис функцій.

- Досліджено правила розробки рекурсивних функцій.
- Навчено розробляти і використовувати на практиці власні функції для вирішення завдань, які стоять перед розробником програм.