Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(ВлГУ)

Институт прикладной математики, физики и информатики

Кафедра физики и прикладной математики

ЛАБОРАТОРНАЯ РАБОТА №2  
по дисциплине  
«Основы программирования»  
на тему:  
«Реализация рекурсивных алгоритмов на С ++»

Выполнил:  
ст. гр. ПМИ-123

Рушев А.М.  
  
  
Принял:  
ст. преподаватель   
каф. ФиПМ  
Шишкина М.В.

Владимир,2024

**Цель работы:**

Изучение принципов работы с функциями в языке программирования С++. Получение навыков объявления, определения, вызова функции, изучение способов передачи параметров в функции, работы с возвращаемым из функции значением.

**Задание**

Опишите следующие функции и приведите несколько примеров её вызова (с различными значениями входных параметров)

1. Инициализации одномерного целочисленного массива.
2. Вывода на экран значений двумерного массива в табличном виде.
3. Вывода на экран столбца таблицы умножение на заданное в качестве параметра число.
4. Вычисление суммы минимальных значений в столбцах двумерного массива.
5. Вычисление суммы максимальных значений в строках двумерного массива.
6. Отсортировки методом выбора одномерного целочисленного массива.
7. Отсортировки методом простой вставки двумерного целочисленного массива.
8. Вычисления факториала заданного числа.
9. Вычисления суммы первых n натуральных чисел, где n – параметр.
10. Вычисления разности квадратов максимального и минимального значений в одномерном целочисленном массиве.

**Теоретическая часть**

Функция - именованная последовательность описаний и операторов, направленных на решение некой задачи. По факту, функция - подпрограмма внутри программы. При использовании функций выделяют три этапа: объявление, определение и вызов:

[<класс памяти>] тип возвращаемого значения> <имя функции> ([список формальных параметров]) {тело функции};

Функция обязательно должна быть объявлена до её вызова. Определение функции, т.е. написание последовательности объявлений и действий, называемых телом функции, может быть совмещено с объявлением функции или вынесено в отдельный модуль.

В любом случае, объявление функции должно содержаться в модуле, в котором она будет вызвана. Определение функции содержит заголовок функции и тело функции – последовательность действий, заключённых в фигурные скобки.

Объявление:

Класс памяти задаёт область видимости функции. Если при объявлении класс памяти не указан, будет использован класс памяти по умолчанию для функций - extern. Функции, объявленные классом памяти extern, доступны для вызова во всех модулях программы.

Функции, объявленные с классом памяти static, доступны для вызова только том модуле, в котором они определены.

Тип возвращаемого значения - тип значения, возвращаемого в точку вызова функции. Кроме того, могут существовать функции, не возвращающие значений. Например,

void f ()

{…}

Указание типа void означает то, что в точку вызова не будут возвращаться значения.

Другой пример

int f (int x, int y) {return x + y;};

Для вызова функции нужно указать имя функции и в круглых скобках список фактических параметров:

int sum (int a, int b) {

return a + b;

}

int main (){

int x = 2, y = 3;

cout << sum (x, y);

}

В данной программе функция сначала вернет сумму фактических параметров x и y в точку вызова, а позже это значение будет выведено и ответ будет равен 5.

Передать параметры в функцию можно тремя способами:

1. по значению;

2. по указателю;

3. по ссылке.

Пример:

int summa (int a, int \*b, int& c)

{

\*b = ++a;

c = a++;

return (a + \*b + c);

}

int main ()

{

int a = 1, \*b = &a, &c = a;

printf ("%i", summa (a, b, c));

}

Параметры по умолчанию получают своё значение уже при объявлении функции. При вызове функции эти параметры можно не указывать. В этом случае будет использовано значение по умолчанию.

Параметры со значением по умолчанию обязательно должны находиться в конце списка формальных параметров. В случае, когда таких параметров несколько и один из низ опущен при вызове, должны быть опущены и все следующие за ним параметры, следовательно, они примут значение по умолчанию. Например.

void f (int a, int b=1, int с=0);

…

f (1);

f (1,2);

**Практическая часть**

**Листинг кода программы:**

#include <iostream>

using namespace std;

const int n = 5;

const int m = 5;

//1 задание

int InitMassiv() {

int arr[5];

for (int i = 0; i < 5; i++) {

arr[i] = 1 + rand() % 50;

cout << arr[i] << "\t";

}

cout << endl;

return \*arr;

}

//2 задание

void PrintMassiv(int n, int m, int\*\* mass) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << mass[i][j] << "\t";

}

cout << endl;

}

}

//3 задание

int TableProizvod(int x) {

int result;

for (int i = 1; i <= 9; i++) {

result = x \* i;

cout << x << "x" << i << " = " << result << endl;

}

return result;

}

//4 задание

int MinSumCellsMassiv(int n, int m, int\*\* mass) {

int sumMinElem = 0;

for (int i = 0; i < n; i++) {

int minElem = mass[i][0];

for (int j = 0; j < m; j++) {

if (minElem > mass[j][i]) {

minElem = mass[j][i];

}

}

sumMinElem += minElem;

}

return sumMinElem;

}

//5 задание

int MaxSumRowsMassiv(int n, int m, int \*\*mass) {

int sumMaxElem = 0;

for (int j = 0; j < m; j++) {

int maxElem = mass[0][j];

for (int i = 0; i < n; i++) {

if (maxElem < mass[j][i]) {

maxElem = mass[j][i];

}

}

sumMaxElem += maxElem;

}

return sumMaxElem;

}

//6 задание

void SortViborMassiv(int n, int \*arr) {

for (int i = 0; i < n; i++) {

int k = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[k]) {

int temp = arr[j];

arr[j] = arr[k];

arr[k] = temp;

}

}

}

}

//7 задание

void SortVstavkaMassiv(int n, int m, int\*\* mass) {

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

int pos = j;

int t = \*(\*(mass + i) + j);

for (int k = j + 1; k < m; k++)

{

if (\*(\*(mass + i) + k) < t)

{

pos = k;

t = \*(\*(mass + i) + k);

}

}

\*(\*(mass + i) + pos) = \*(\*(mass + i) + j);

\*(\*(mass + i) + j) = t;

}

}

}

//8 задание

int Factorial(int \_x) {

int F = \_x;

for (int i = 1; i < \_x; i++) {

F \*= i;

}

return F;

}

//9 задание

int SumNatural(int y) {

int sum = 0;

for (int i = 1; i < y; i++) {

sum += i;

}

return sum;

}

//10 задание

int DifferenceOfSquares(int \*arr, int n) {

int result;

int maxElem = arr[0];

int minElem = arr[0];

for (int i = 0; i < n; i++) {

if (arr[i] > maxElem) {

maxElem = arr[i];

}

if (arr[i] < minElem) {

minElem = arr[i];

}

}

result = (maxElem \* maxElem) - (minElem \* minElem);

return result;

}

int main()

{

setlocale(LC\_ALL, "rus");

//1 задание

cout << "Задание 1." << endl;

InitMassiv();

//2 задание

cout << "\nЗадание 2." << endl;

//Создание массива

int\*\* mass = new int\* [m];

for (int i = 0; i < m; i++) {

mass[i] = new int[n];

}

//Заполнение массива

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

mass[i][j] = 1 + rand() % 50;

}

}

//Вызов функции вывода массива

PrintMassiv(n, m, mass);

//Задание 3

cout << "\nЗадание 3." << endl;

int x;

cout << "Введите число: ";

cin >> x;

TableProizvod(x);

//Задание 4

cout << "\nЗадание 4." << endl;

int sumCellsMin = MinSumCellsMassiv(n, m, mass);

cout << "Сумма минимальных элементов в столбцах массива = " << sumCellsMin << endl;

//Задание 5

cout << "\nЗадание 5." << endl;

int sumRowsMax = MaxSumRowsMassiv(n, m, mass);

cout << "Сумма максимальных элементов в строках массива = " << sumRowsMax << endl;

//Задание 6

cout << "\nЗадание 6." << endl;

int arr[n] = { 5,14,23,6,9 };

cout << "Исходный массив: ";

for (int i = 0; i < n; i++) {

cout << arr[i] << "\t";

}

cout << endl;

SortViborMassiv(n, arr);

cout << "Отсортированный массив: ";

for (int i = 0; i < n; i++) {

cout << arr[i] << "\t";

}

cout << endl;

//Задание 7

cout << "\nЗадание 7." << endl;

SortVstavkaMassiv(n, m, mass);

cout << "Отсортированный массив." << endl;

PrintMassiv(n, m, mass);

//Задание 8

cout << "\nЗадание 8." << endl;

int \_x;

cout << "Введите число для факторила: ";

cin >> \_x;

cout << "Факториал числа " << \_x << " = " << Factorial(\_x);

//Задание 9

cout << "\nЗадание 9." << endl;

int y;

cout << "Введите параметр: ";

cin >> y;

int resNatural = SumNatural(y);

cout << "Сумма натуральных чисел = " << resNatural << endl;

//Задание 10

cout << "\nЗадание 10." << endl;

int resDifferenceOfSquares = DifferenceOfSquares(arr, n);

cout << "Квадрат разности = " << resDifferenceOfSquares << endl;

//Освобождение памяти у массива

for (int i = 0; i < m; i++) {

delete[] mass[i];

}

delete[] mass;

cout << endl;

system("pause");

}

**Результат выполнения кода программы:**

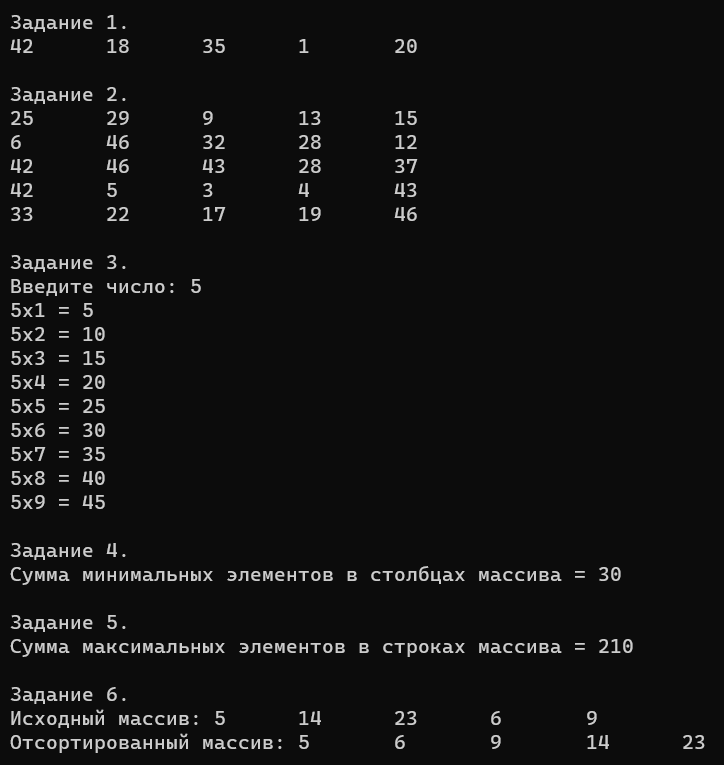


Рисунок 1 – 1 часть работы программы

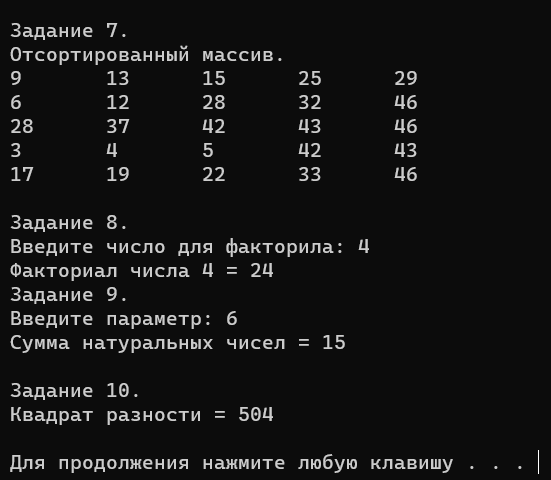


Рисунок 2 – 2 часть выполнения кода программы

**Вывод:**

В ходе лабораторной работы были получены навыки объявления, определения, вызова функции, изучение способов передачи параметров в функции, работы с возвращаемым из функции значением.