Лабораторная работа №7. Функции

1 Цель и порядок работы

Цель работы – изучить возможности языка по организации функций, получить практические навыки в составлении программ с их использованием.

Порядок выполнения работы:
- ознакомиться с описанием лабораторной работы;
- получить задание у преподавателя, согласно своему варианту;
- написать программу и отладить ее на ЭВМ.

2 Краткая теория

Функция – самостоятельная единица программы, спроектированная для реализации конкретной задачи, представляющая именованную группу операторов и выполняющая законченное действие. Вызов функции приводит к выполнению некоторых действий. К функции можно обратиться по имени, передать ей значения и получить из нее результат. Функции избавляют нас от повторного программирования, позволяют использовать их в различных программах, повышают уровень модульности программы. Функции нужны для упрощения структуры программы. Разбив задачу на подзадачи и оформив каждую из них в виде функций, мы улучшаем понятность и структурированность программы.

2.1 Описание функций

Функция может принимать параметры и возвращать значение. Передача в функцию различных аргументов позволяет, записав ее один раз, использовать многократно для разных данных.

Для использования функции требуется знать только ее интерфейс (т. е. правила обращения). Интерфейс грамотно написанной функции определяется ее заголовком, потому что в нем указывается все, что необходимо для ее вызова: имя функции, тип результата, который она возвращает, а также, сколько аргументов и какого типа ей нужно передать.

Объявление функции (прототип, заголовок, сигнатура) задает ее имя, тип возвращаемого значения и список передаваемых параметров. Формат заголовка (прототипа) функции:

```
[ класс ] тип имя ([ список_формальных_параметров ]);
```

Определение функции содержит, кроме объявления, тело функции, представляющее собой последовательность операторов и описаний в фигурных скобках:

```
[ класс ] тип имя ([ список_формальных_параметров ])
{
   тело функции
}
```

Функция активируется с помощью оператора вызова функции, в котором содержатся имя функции и параметры (если это необходимо). Вызов функции приводит к выполнению операторов, составляющих тело функции, и выглядит следующим образом:

```
имя ([ список_фактических_параметров ]);
```

Функцию можно определить как встроенную с помощью модификатора **inline**, который рекомендует компилятору вместо обращения к функции помещать ее код непосредственно в каждую точку вызова. Модификатор **inline** ставится перед типом функции.

Все величины, описанные внутри функции, а также ее параметры, являются локальными. Областью их действия является функция. При вызове функции, как и при входе в любой блок, в стеке выделяется память под локальные автоматические переменные. При выходе из функции соответствующий участок стека освобождается, поэтому значения локальных переменных между вызовами одной и той же функции не сохраняются. Если этого требуется избежать, при объявлении локальных переменных используется модификатор **static**.

Возврат вычисленного значения организуется следующим образом. В теле функции должен присутствовать оператор **return** после которого следует выражение, вычисляющее возвращаемое значение. Таких операторов может быть несколько; важно, чтобы хоть один из них срабатывал в процессе выполнения тела функции. Тип выражения в правой части такого присваивания должен быть совместимым с типом функции.

Пример функции, возвращающей сумму двух целых величин:

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int sum(int a, int b); // объявление функции

void main()
{

   int a = 2, b = 3, c, d;
   c = sum(a, b);              // вызов функции
   cin >> d;
   cout << sum(c, d) << endl; // вызов функции

   a = 3*sum(c, d) - b;        // вызов функции
   cout << a << endl;
   return;
}

int sum(int a, int b) // определение функции
{
   return (a + b);
}
```

2.2 Параметры функции

Механизм параметров является основным способом обмена информацией между вызываемой и вызывающей функциями.

Параметры, перечисленные в заголовке описания функции, называются формальными параметрами, или просто параметрами, а записанные в операторе вызова функции – фактическими параметрами, или аргументами.

Существует два основных способа передачи параметров в функцию: **по значению** и **по адресу**.

При передаче по значению в стек заносятся копии значений аргументов, и операторы функции работают с этими копиями. Доступа к исходным значениям параметров у функции нет, а, следовательно, нет и возможности их изменить.

При передаче по адресу в стек заносятся копии адресов аргументов, а функция осуществляет доступ к ячейкам памяти по этим адресам и может изменить исходные значения аргументов. Данный способ подразделяется на передачу параметров через указатель и по ссылке.

Если требуется изменить значение параметра внутри функции, то он передается, либо через ссылку, либо через указатель.

Если требуется запретить изменение параметра внутри функции, используется модификатор **const**:

```
int f(const char *);
```

По умолчанию параметры любого типа, кроме массива и функции (например, вещественного, структурного, перечисление, объединение, указатель), передаются в функцию по значению.

Если в определении функции присутствует параметр со значением по умолчанию, то данный параметр можно опустить при вызове функции. При этом все параметры левее данного должны присутствовать. Значение данного параметра внутри функции будет совпадать со значением параметра по умолчанию.

Пример: Составить программу вычисления значения функции

$$a^x = 1 + \sum_{n=1}^{\infty} \frac{(x \ln a)^n}{n!} = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \dots$$

```
#include "stdafx.h"
#include <iostream>
#include <iomanip>
#include <math.h>

using namespace std;

float ax(float a, float x, float eps = 0.001f)
{

   float xn = 1, y, y0;

   unsigned int n = 1, nf = 1;

   y = 1;

   do
   {
       nf *= n;
       xn *= x*log(a);
       y0 = y;
       y += xn/nf;
       n++;
   }
   while (abs(y - y0) > eps);

   return y;
}

void main()
{
   setlocale(LC_ALL, "Russian");

   const float eps = 0.1f;
```

```
    float a, x, y;

    cout << "Введите a, x:" << endl;
    cin >> a >> x;

    //передаем значение точности вычислений при вызове
    y = ax(a, x, eps);

    cout <<\
"Результат вычисления с точностью 0.1 (передается явно): "\
<< setw(8) << setprecision(5) << y <<endl;

//значение esp опускается
//внутри функции используется значение точности вычислений по умолчанию
    y = ax(a, x);

    cout <<\
"Результат вычисления с точностью 0.001 (по умолчанию): "\
<< setw(8) << setprecision(5) << y <<endl;

    //вычисляем при помощи стандартных функций
    y = pow(a, x);

    cout <<\
"Результат вычисления с использованием стандартных функций: "\
<< setw(8) << setprecision(5) << y <<endl;

}
```

Результат выполнения:

```
Введите a, x:
2.5
3
Результат вычисления с точностью 0.1 (передается явно):    15.591
Результат вычисления с точностью 0.001 (по умолчанию):     15.625
Результат вычисления с использованием стандартных функций: 15.625
```

2.2 Передача массивов в функцию

Параметрами функции могут быть массивы, и функции могут возвращать указатель на массив в качестве результата. При использовании массивов в качестве параметров в функцию передается указатель на его первый элемент, т.е. массив всегда передается по адресу.

Если аргументом функции является одномерный массив, ее формальный параметр можно объявить тремя способами: как указатель, как массив фиксированного размера и как массив неопределенного размера.

Если двухмерный массив используется в качестве аргумента функции, то в нее передается только указатель на его первый элемент. При этом необходимо указать количество столбцов. Количество столбцов необходимо компилятору для того, чтобы правильно вычислить адрес элемента массива внутри функции.

2.3 Рекурсивные функции

Любая функция в программе на C++ может вызываться рекурсивно. При этом в стеке выделяется новый участок памяти для размещения копий параметров, а также автоматических и регистровых переменных, поэтому предыдущее состояние выполняемой функции сохраняется, и к нему впоследствии можно вернуться.

Различают прямую и косвенную рекурсию. Функция называется косвенно-рекурсивной в том случае, если она содержит обращение к другой функции, содержащей прямой или косвенный вызов определяемой первой функции. Если в теле функции используется явный вызов этой же функции, то это прямая рекурсия.

Однако у рекурсии есть и недостатки: во-первых, такую программу труднее отлаживать, поскольку требуется контролировать глубину рекурсивного обращения, во-вторых, при большой глубине стек может переполниться, а в-третьих, использование рекурсии увеличивает расход времени и памяти на повторные вызовы функции.

3 Контрольные вопросы

1. Как выглядит определение функции?
2. Как выглядит объявление функции?
3. Что такое формальный параметр?
4. Что такое фактический параметр?
5. Как осуществляется вызов функции?
6. Как осуществляется передача параметров в функцию?
7. Что такое `inline`-функция?
8. Как описать функцию, не возвращающую значения?
9. В чем разница передачи параметров по значению от передачи параметров по адресу?
10. Как передать параметр по ссылке?
11. Как задать значения параметра по умолчанию?
12. Как передать массив в функцию?
13. Как передать многомерный массив в функцию?
14. Что такое рекурсия?
15. Какие виды рекурсии вы знаете?

4 Задание

1. Написать программу в соответствии с вариантом задания из пункта 5.1.
2. Отладить и протестировать программу.
   Написать программу в соответствии с вариантом задания из пункта 5.2.
3. Отладить и протестировать программу.
4. Написать программу в соответствии с вариантом задания из пункта 5.3.
5. Отладить и протестировать программу.
6. Написать программу в соответствии с вариантом задания из пункта 5.4.
7. Отладить и протестировать программу.

5 Варианты заданий

5.1 Функции, параметры функций

Варианты заданий выбирать согласно номера в списке группы.

Определить три функции, выполняющие действия в соответствии с вариантом задания, по одной на каждый способ передачи параметров. Написать программу, осуществляющую вызов этих функций несколько раз с различными параметрами.

1. Вычислить с использованием подпрограммы – функции Z=НОД(a,b)+НОК(a,b), где a, b – целые положительные числа, НОД – наибольший общий делитель, НОК – наименьшее общее кратное.
2. Определить функцию нахождения расстояния между точками. Во множестве точек на плоскости найти пару точек с максимальным расстоянием между ними.
3. Найти наибольшую из высот треугольника. Известны две стороны треугольника и угол между ними.
4. Найти: y=среднее(a,b,c)/min(a,b,c).
5. Даны действительные числа s,t. Получить g(1.2,s) + g(t,s) - g(2s-1,st), где
$$g(a,b)=\frac{a^2-b^2}{2\cdot a\cdot b-a-b}+(a+b)\cdot\sqrt{\frac{|a+b|}{2}}$$
6. Вычислить сумму значений функций
$$Z=f(\sin(x)+\cos(y),x+y,2)+f(x\cdot y,\sin(x),\cos(y))+f(\sin^2(x),x-y^2,y-x^2)$$
где
$$f(u,v,t)=\begin{cases} v\cdot u\cdot t, & \text{если } u>1\\ u+t+v, & \text{если } 0\le u\le 1\\ v-t-u, & \text{если } 0<u\end{cases}$$
7. .
8. Даны действительные числа s,t. Получить g(1.2,s) + g(t,s) - g(2s-1,st), где
$$g(a,b)=\frac{a^2+b^2-4\cdot a\cdot b}{a^2+5\cdot a\cdot b+3\cdot b^2+4\cdot a-b}$$
9. Составить программу вычисления суммы квадратов простых чисел, лежащих в интервале [M, N].
10. Даны отрезки a, в, c и d. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. (Определить функцию, вычисляющую площадь треугольника, если она существует)
11. Определить функцию нахождения расстояния между точками. Во множестве точек на плоскости найти пару точек с минимальным расстоянием между ними.
12. Найти: y=min(a,b,c)/max(a,b,c).
13. Вычислить сумму значений функций
$$Z=f(\sin(x)+\cos(y),x+y)+f(\sin(x),\cos(y))+f(\sin^2(x)-2,a+b^2)$$
Где
$$f(u,t)=\begin{cases} u+t, & \text{если } u>1\\ u-t, & \text{если } 0\le u\le 1\\ t-u, & \text{если } 0<u\end{cases}$$
14. Даны значения a и b, найти их среднее арифметическое, среднегеометрическое.
15. Найти: y=max(a,b,c)+min(a,b,c).

16. Вычислить с использованием подпрограммы – функции Z=НОК(a+b,a*b)+НОК(a, b), где a, b – целые положительные числа, НОД – наибольший общий делитель, НОК – наименьшее общее кратное.
17. Вычислить среднее геометрическое шести вводимых чисел.
18. Вычислить сумму значений функции Z=F(a,b)+F(a2,b2)+F(a2-1,b)+F(a-b,b)

$$F(u,t)=\begin{cases} u^2+t^2, & \text{если} \quad u>0, \quad t>0 \\ u+t^2, & \text{если} \quad u\le 0, \quad t\le 0 \\ u-t, & \text{если} \quad u>0, \quad t\le 0 \\ u+t, & \text{если} \quad u\le 0, \quad t>0 \end{cases}$$

19. Даны действительные числа s, t. Получить f(t, -2s, 1.17) + f(2.2,t,s-t), где

$$f(a,b,c)=\frac{2\cdot a-b-\sin(c)+a\cdot b}{1+|c+a|}$$

20. Найти: y=max(a,b,c,d)*min(a,b,c,d).
21. Вычислить с использованием процедуры $Z=\dfrac{\sum_{i=1}^{40}\sin(x_i)+\sum_{i=1}^{40}\cos(y_i)}{\sum_{i=1}^{40}|x_i|}$. Каждую сумму вычислять с использованием одной подпрограммы.
22. Заданно множество точек на плоскости. Найти сумму длин отрезков между ними.
23. Вычислить с использованием функции Z=НОД(a,b)+НОД(a*b,a+b), где a, b – целые положительные числа, НОД – наибольший общий делитель.
24. Вычислить сумму значений функции

$$Z=f(\sqrt{|x|},y)+f(a,b)+f(\sqrt{|x|}+1,-y)+f((|x|-|y|),x),$$

$$\text{где} \quad f(u,t)=\begin{cases} u+2t, & \text{если} \quad u\ge 0 \\ u+t, & \text{если} \quad u\le -1 \\ u^2-2t+1, & \text{если} \quad -1<u<0 \end{cases}$$

25. Вычислить среднее арифметическое четырех вводимых чисел.

5.2 Передача массивов в функцию (одномерные массивы)

Варианты заданий выбирать согласно номера в списке группы.

Определить функции, выполняющие действия в соответствии с вариантом задания.

1. Дан одномерный массив, состоящий из N целочисленных элементов.
   - 1.1. Найти максимальный положительный элемент.
   - 1.2. Вычислить сумму элементов массива.
2. Дан одномерный массив, состоящий из N вещественных элементов.
   - 2.1. Найти максимальный элемент.
   - 2.2. Вычислить среднеарифметическое отрицательных элементов массива.
3. Дан одномерный массив, состоящий из N вещественных элементов.
   - 3.1. Найти минимальный элемент.
   - 3.2. Вычислить произведение не нулевых элементов массива.
4. Дан одномерный массив, состоящий из N целочисленных элементов.
   - 4.1. Найти минимальный положительный элемент.
   - 4.2. Вычислить сумму положительных элементов массива, кратных 3.
5. Дан одномерный массив, состоящий из N целочисленных элементов.

5.1. Найти максимальный положительный элемент.
5.2. Вычислить произведение элементов массива.

6. Дан одномерный массив, состоящий из N вещественных элементов.
6.1. Найти максимальный элемент.
6.2. Вычислить сумму четных элементов массива.

7. Дан одномерный массив, состоящий из N вещественных элементов.
7.1. Найти минимальный отрицательный элемент.
7.2. Вычислить среднеарифметическое положительных элементов массива.

8. Дан одномерный массив, состоящий из N целочисленных элементов.
8.1. Найти максимальный элемент.
8.2. Вычислить среднеарифметическое элементов массива.

9. Дан одномерный массив, состоящий из N целочисленных элементов.
9.1. Найти минимальный элемент.
9.2. Вычислить сумму элементов массива.

10. Дан одномерный массив, состоящий из N целочисленных элементов.
10.1. Найти максимальный отрицательный элемент.
10.2. Вычислить произведение отрицательных элементов массива.

11. Дан одномерный массив, состоящий из N целочисленных элементов.
11.1. Найти максимальный элемент.
11.2. Вычислить среднеарифметическое нечетных элементов массива.

12. Дан одномерный массив, состоящий из N вещественных элементов.
12.1. Найти минимальный положительный элемент.
12.2. Вычислить сумму четных элементов массива.

13. Дан одномерный массив, состоящий из N вещественных элементов.
13.1. Найти минимальный отрицательный элемент.
13.2. Вычислить произведение ненулевых элементов массива, кратных 3.

14. Дан одномерный массив, состоящий из N целочисленных элементов.
14.1. Найти максимальный отрицательный элемент.
14.2. Вычислить среднеарифметическое четных элементов массива.

15. Дан одномерный массив, состоящий из N вещественных элементов.
15.1. Найти максимальный элемент.
15.2. Вычислить среднеарифметическое положительных элементов массива.

16. Дан одномерный массив, состоящий из N вещественных элементов.
16.1. Найти минимальный положительный элемент.
16.2. Вычислить произведение не нулевых элементов массива.

17. Дан одномерный массив, состоящий из N целочисленных элементов.
17.1. Найти максимальный отрицательный элемент.
17.2. Вычислить сумму отрицательных элементов массива.

18. Дан одномерный массив, состоящий из N целочисленных элементов.
18.1. Найти минимальный элемент.
18.2. Вычислить сумму положительных нечетных элементов массива.

19. Дан одномерный массив, состоящий из N вещественных элементов.
19.1. Найти минимальный положительный элемент.
19.2. Вычислить произведение нечетных элементов массива.

20. Дан одномерный массив, состоящий из N вещественных элементов.
20.1. Найти максимальный элемент.
20.2. Вычислить среднеарифметическое отрицательных элементов массива.

21. Дан одномерный массив, состоящий из N целочисленных элементов.
21.1. Найти максимальный положительный элемент.
21.2. Вычислить сумму положительных четных элементов массива.

22. Дан одномерный массив, состоящий из N целочисленных элементов.
22.1. Найти минимальный элемент.

22.2. Вычислить произведение ненулевых нечетных элементов массива.

23. Дан одномерный массив, состоящий из N вещественных элементов.
    23.1. Найти минимальный положительный элемент.
    23.2. Вычислить среднеарифметическое положительных элементов массива.
24. Дан одномерный массив, состоящий из N вещественных элементов.
    24.1. Найти максимальный отрицательный элемент.
    24.2. Вычислить среднеарифметическое нечетных элементов массива.
25. Дан одномерный массив, состоящий из N целочисленных элементов.
    25.1. Найти минимальный отрицательный элемент.
    25.2. Вычислить сумму несчетных отрицательных элементов массива.

5.3 Передача массивов в функцию (многомерные массивы)

Написать программу, выполняющую действия в соответствии с вариантом задания и передающую массив в функцию. Ввод и вывод массивов выполнить в отдельных функциях.
Вариант определяется по последней цифре в списке студентов группы.

1. Вычислить с использованием функции наименьшие элементы в строке и сумму номеров строк и столбцов, в которых они расположены, для матрицы A(10,15). Результаты формировать в одномерных массивах M(10) и S(10).
2. Дан массив a(8,5). С использованием функции найти среднеквадратичное значение положительных элементов каждой строки массива и сформировать из них одномерный массив b(8).
3. Вычислить с использованием функции max элементы каждой строки матрицы A(10,20). Результаты формировать в одномерных массивах C(10) и D(10).
4. Даны массивы a(3,4), b(2,5). Найти Z = (Ma+Mb)/(da+db), где Ma ,Mb - среднеарифметические значения массивов A, B. da, db - максимальные отклонения от среднеарифметических значений.
5. Дана матрица A(5,5). Сформировать одномерный массив C(5) из среднегеометрических значений положительных элементов каждого столбца матрицы.
6. Вычислить $Z = \frac{x_{\max} - y_{\min}}{x_{\min} - y_{\max}}$ с использованием функции, где xmax ,xmin ,ymax ,ymin – максимальные и минимальные элементы соответственно массива x(5,2) и массива y(3,4).
7. Дана матрица A(4,5), B(5,6). Вычислить $Z = p_a + p_b, \quad \text{где} \quad p = \sum_{i=1}^{N} \max\{x_{ij}\} \quad j = 1, N$ сумма максимальных элементов каждой строки матрицы.
8. Вычислить с использованием функции min элементы каждой строки матрицы A(10,20). Результаты формировать в одномерных массивах C(10) и D(10).
9. Преобразовать массив x(3,3) в y, оставив в нем только положительные элементы. Вместо остальных элементов записать 0.
10. Определить количество положительных, отрицательных и нулевых элементов матрицы A(10,15). (Создать три функции для нахождения этих значений).

5.4 Использование рекурсии

Написать программу, рекурсивно вычисляющую сумму.
Вариант определяется по последней цифре в списке студентов группы.

1. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{n!}{5^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
2. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{1}{2^n} + \frac{1}{3^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
3. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{(-1)^{n-1}}{n^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
4. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{1}{(3n-2)(3n+1)}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
5. Найти сумму ряда с точностью ε, общий член которого равен $a_n = n^2 e^{-\sqrt{n}}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
6. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{n!}{2n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
7. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{3 \cdot n!}{(2n)!}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
8. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{\ln(n!)}{n^2}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
9. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{(2n-1)}{2^n}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.
10. Найти сумму ряда с точностью ε, общий член которого равен $a_n = \frac{10^n}{n!}$. Точность считается достигнутой, если следующий член последовательности меньше заданного ε.