

Практическая работа №12

Функции

1. Задание

Составить программу, которая вычисляет значение функции трех переменных $R(x,y,z)$ указанной в варианте задания (см. табл. 1) для заданных значениях параметров.

2. Требования к выполнению задания

При разработке программы необходимо придерживаться следующих основных требований:

- 1) значения параметров, указанных в варианте задания, следует вводить с клавиатуры.
- 2) вычисление указанных в варианте задания функций должно выполняться для произвольных значений порядка $n \geq 1$ и аргумента x с помощью соответствующих подпрограмм-функций по правилам, указанным в табл. 2;
- 3) для вычисления значений функции следует применять циклы, основанные на рекуррентных соотношениях;
- 4) значения порядка функции n и аргумента x должны передаваться подпрограммам в качестве параметров;
- 5) подпрограммы должны корректно выполняться при любом значении порядка $n \geq 1$;
- 6) обязательно наличие исходного текста программы, записанного в файле;
- 7) все операторы должны оформляться с помощью отступов.

3. Примечания

3.1 Понятие подпрограммы

Подпрограммой называется автономная, специальным образом оформленная часть программы, обладающая собственной (локальной) памятью. Подпрограммы являются средством языка программирования, обеспечивающим поддержку следующих основных функций:

- оформление повторяющихся участков программы, позволяющее многократно использовать один и тот же текст с различными исходными данными без его перезаписи;
- упрощение разработки больших программ путем разбиения их на части, обладающие высокой степенью автономности, соответствующие отдельным подзадачам решаемой задачи;
- оформление часто используемых алгоритмов, позволяющее включать их состав нескольких программ без перезаписи текста.

Основным свойством подпрограмм, которое обеспечивает перечисленную выше функциональность, является свойство локальности.

3.2 Локальность

Свойство локальности выражается в следующих аспектах функционирования подпрограмм:

- обрабатываемые данные и выполняемые подпрограммой действия локализуются внутри подпрограммы;
- локализация данных означает, что они могут обрабатываться только операторами подпрограммы, а обращение к локальным данным из других частей программы невозможно;
- между вызывающей программой и подпрограммой необходимо организовать обмен данными, который реализуется с помощью параметров подпрограммы;
- локализация действий подразумевает, что выполнение операторов подпрограммы не может произвольным образом чередоваться с выполнением операторов других частей программы;

- для выполнения действий, предусмотренных в подпрограмме, требуется выполнение специальной процедуры вызывающей передачу управления этой подпрограмме от вызывающей программы (вызов подпрограммы);
- передача управления требует приостановить выполнение операторов вызывающей программы и выполнить операторы, входящие в состав подпрограммы;
- после завершения подпрограммы управление возвращается вызывающей программе, т.е. продолжается ее выполнение с места приостановки.

3.3 Объявление функций

Все функции, которые используются в программе, должны быть объявлены до их первого использования.

Для объявления функции можно определить ее прототип.

Прототип функции ::= [класс] <тип> <имя> (<список формальных параметров>);

Необязательный модификатор класс явно задает область видимости функции и может принимать значения `extern` (по умолчанию) и `static`. Значение `extern` определяет глобальную (для всех модулей) видимость функции, а `static` задает видимость только в пределах модуля. Прототип функции задаёт имя функции, тип возвращаемого значения, а также количество и типы параметров, которые необходимо указать при вызове функции. Функция может возвращать любой тип данных, за исключением массивов. Указание `void` в качестве типа возвращаемого значения означает, что функция не возвращает значения.

Список формальных параметров – это список, определяющий исходные данные для работы функции. Элементы списка отделяются друг от друга запятыми.

Список формальных параметров ::=

[<тип> <имя параметра>] [,<тип> <имя параметра>...]

Каждый элемент списка состоит из идентификатора типа и имени параметра. При вызове функции для параметров следует указать исходные значения. Тип возвращаемого значения и типы параметров совместно определяют тип функции.

Функция может быть и без параметров, тогда их список будет пустым. Такой пустой список можно указать в явном виде, поместив для этого внутри скобок ключевое слово `void`.

Прототип функции недостаточен для ее использования в программе, поскольку в нем не задаются действия, выполняемые функцией. Для того чтобы задать выполняемые действия, используется определение функции. Определением функции является объявление функции, за которым следует составная инструкция (тело функции).

Определение функции:

```
[класс] <тип> <имя> (<список формальных параметров>)
{
    [<объявления>]
    [<операторы>]
}
```

Типы в определении и объявлениях функции должны совпадать. Однако имена параметров не являются частью типа и не обязаны совпадать.

Все функции в программе не могут быть вложены друг в друга.

В программе всегда должна присутствовать единственная функция, которая называется главной и должна иметь имя `main`. С нее начинается выполнение программы, обычно она управляет выполнением программы, организуя вызовы других функций. Для того чтобы программа могла быть скомпилирована и выполнена, она должна содержать, по крайней мере, определение функции `main`.

Прототипы функций следует размещать до определения главной функции. Определение функции также можно использовать в качестве ее прототипа, в этом случае определение также размещается перед определением главной функции.

3.4 Оператор return

Этот оператор имеет два важных применения:

Во-первых, он обеспечивает немедленный выход из функции, т.е. заставляет выполняющуюся программу передать управление коду, вызвавшему функцию в точку, непосредственно следующую за вызовом.

Во-вторых, этот оператор можно использовать для того, чтобы вернуть значение функции.

Все функции, кроме тех, которые относятся к типу void, возвращают значение. Это значение указывается выражением в операторе return. Синтаксис оператора return задается следующей конструкцией:

Оператор return ::= return [<выражение>]

3.5 Вызов функции

Для выполнения действий, определенных в функции, необходимо осуществить ее вызов. Вызов функции выполняется путем указания ее имени, за которым следует список выражений, заключенный в круглые скобки (список фактических аргументов).

Вызов функции ::= <имя> ([<выражение>] [,<выражение>...])

Если функция не объявлена как имеющая тип void, она может использоваться как операнд в выражении.

В определении, объявлении и при вызове одной и той же функции типы и порядок следования параметров должны совпадать. На имена параметров ограничений по соответствию не накладывается, поскольку функцию можно вызывать с различными аргументами, а в прототипах имена компилятором игнорируются (они служат только для улучшения читаемости программы).

3.6 Пример программы

Числовая последовательность определена рекуррентной формулой:

$$A_{i+1} = \frac{2i+1}{A_i} + \frac{A_{i-1}}{i}; A_1 = 1,2; A_2 = 2,3.$$

Составить подпрограмму, вычисляющую значение элемента последовательности для произвольного заданного номера этого элемента.

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

```
double A(int n);           // Прототип функции A.
```

```
int main()  
{  
    double An;  
    int n;  
    scanf("%i",&n);  
    An=A(n);  
    printf("An= %f\n",An);  
    system("pause");  
}
```

```
double A(int n) {  
    double a1,a2,a3;
```

```

int i;
a2=1.2;
a3=2.3;
switch (n) {
    case 1: return a2;
    case 2: return a3;
    default:
        for (i=2;i<=n-1;i++) {
            a1=a2;
            a2=a3;
            a3=(2*i+1)/a2-a1/i;
        }
    return a3;
}
}

```

4. Пример выполнения задания

Задание

Заданы следующие функции:

$$f_1(x) = \frac{\sin(x)}{x}, \quad f_2(y) = y^2 + 2y^{3/2}, \quad f_3(z) = (z+2)(z+1)z.$$

Вычислить значение функции

$$R(x, y, z) = f_1(x) + f_2(y)f_3(f_2(z)) \text{ для } x = 0.6, \quad y = 0.3, \quad z = 0.1.$$

Тесты

$$f_1(0) = 1, \quad f_1(0.5) = 0.959, \quad f_1(1) = 0.841;$$

$$f_2(0) = 0, \quad f_2(0.5) = 0.957, \quad f_2(1) = 3;$$

$$f_3(0) = 0, \quad f_3(0.5) = 1.875, \quad f_3(1) = 6;$$

$$R(0,0,0) = 1, \quad R(1,1,1) = 180.841, \quad R(0.5,0.5,0.5) = 6.260.$$

Программа

```

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

double f1(double x);
double f2(double y);
double f3(double z);

int main()
{
    double x,y,z,ff1,ff2,ff3,R;
    scanf("%lf %lf %lf",&x,&y,&z);
    ff1=f1(x);
    ff2=f2(y);
    ff3=f3(z);
    printf("f1= %f f2= %f f3= %f\n",ff1,ff2,ff3);
    R=f1(x)+f2(y)*f3(f2(z));
    printf("R= %f\n",R);
    system("pause");
}

```

```

}
```

```

double f1(double x) {
    double f;
    if (x!=0)
        f=sin(x)/x;
    else
        f=1;
    return f;
}

```

```

double f2(double y) {
    double f;
    f=y*y+2*pow(y,1.5);
    return f;
}

```

```

double f3(double z){
    double f;
    f=(z+2)*(z+1)*z;
    return f;
}

```

5. Варианты заданий

Таблица 1. Определение функции $F(x, y, z)$

№ п/п	Функция $F(x, y, z)$	x	y	z	Ожидаемый результат
1.	$A_8(x) + K_1(L_5(y) + A_9(z))$	0,3	0,5	0,1	-0,63602
2.	$D_6(x)J_1(B_2(y) + B_4(z))$	0,5	0,4	0,1	0,35771
3.	$C_3(x) - E_6(F_2(y)) + F_5(z)$	0,3	0,3	0,2	-2,26462
4.	$E_1(x)F_7(G_2(y)G_5(z))$	0,1	0,1	0,4	-0,00805
5.	$K_5(x)L_7(H_8(y)L_2(z))$	0,4	0,2	0,5	-0,29111
6.	$I_5(x) + J_7(H_8(y) - J_2(z))$	0,3	0,1	0,4	0,00139
7.	$D_7(x) - L_2(H_4(y))L_9(z)$	0,2	0,3	0,1	0,31221
8.	$F_3(x) - G_6(I_8(y)) + I_1(z)$	0,4	0,2	0,1	-0,31667
9.	$H_5(x)J_4(A_2(y) + A_4(z))$	0,4	0,5	0,3	0,32220
10.	$B_1(x) + E_9(L_5(y) + E_2(z))$	0,3	0,2	0,4	0,09023
11.	$J_8(x) + E_1(E_5(y) + D_9(z))$	0,3	0,5	0,1	-0,21824
12.	$A_2(x) + K_1(C_3(y) - C_7(z))$	0,5	0,3	0,5	-1,22646
13.	$G_2(x) - G_7(K_7(y)) + L_{10}(z)$	0,4	0,2	0,3	-0,20015
14.	$E_8(H_7(x))L_1(y) + L_7(z)$	0,5	0,3	0,4	1,48397
15.	$I_9(J_4(x))K_4(y) - J_5(z)$	0,2	0,2	0,1	-0,16038
16.	$D_9(B_5(x) + D_2(y)) - A_{10}(z)$	0,2	0,4	0,1	-0,84815
17.	$F_7(x) - G_5(I_4(y))I_9(z)$	0,1	0,2	0,5	-0,00042
18.	$C_7(x) + J_7(K_5(y) - J_2(z))$	0,4	0,2	0,4	-2,69106
19.	$A_6(x)B_7(C_2(y)B_3(z))$	0,5	0,5	0,1	-0,00023
20.	$B_7(x) - C_7(D_2(y)) + D_5(z)$	0,2	0,5	0,3	6,87713

21.	$F_8(x)G_7(I_2(y) - F_3(z))$	0,3	0,4	0,4	-0,01904
22.	$D_7(x) - L_2(H_4(y))L_9(z)$	0,5	0,1	0,4	6,42221
23.	$A_5(x) + E_2(I_3(y) - E_7(z))$	0,4	0,2	0,1	1,20857
24.	$E_9(x) + F_7(y)I_8(F_2(z))$	0,1	0,5	0,1	0,17318
25.	$F_3(x) - G_5(H_9(y)) + F_5(z)$	0,2	0,2	0,1	-2,42549

Таблица 2. Правила вычисления функций, указанных в вариантах задания

№ п/п	Функция
1.	$A_{n+1}(x) = \frac{2n+1}{n+1}xA_n(x) - \frac{n}{n+1}A_{n-1}(x), A_1(x) = x, A_2(x) = \frac{1}{2}(3x^2 - 1)$
2.	$B_{n+1}(x) = 2xB_n(x) - \frac{1}{n^2+1}B_{n-1}(x), B_1(x) = 1, B_2(x) = x.$
3.	$C_{n+1}(x) = 2xC_n(x) - \frac{2n}{n+1}C_{n-1}(x), C_1(x) = 2x, C_2(x) = 4x^2 - 2$
4.	$D_{n+1}(x) = \frac{(2n+1-x)}{n}D_n(x) - \frac{n}{n^2+1}D_{n-1}(x), D_1(x) = x, D_2(x) = x^2$
5.	$E_{n+1}(x) = \frac{x}{n+1}E_n(x) - E_{n-1}(x), E_1(x) = x, E_2(x) = x^2 - 1$
6.	$F_{n+1}(x) = \frac{(2n+1)}{n^2}F_n(x) - \frac{n}{n+1}xF_{n-1}(x), F_1(x) = 1, F_2(x) = x^2 - 1$
7.	$G_{n+1}(x) = 2x^2G_n(x) + \frac{n}{n+2}G_{n-1}(x), G_1(x) = 2 - x, G_2(x) = x^2.$
8.	$H_{n+1}(x) = 2xH_n(x) - nx^2H_{n-1}(x), H_1(x) = x, H_2(x) = x^2 - 1$
9.	$I_{n+1}(x) = \frac{(2n-1)}{n^2}I_n(x) - \frac{n^2}{n^3+1}xI_{n-1}(x), I_1(x) = 1, I_2(x) = x^2 - 2$
10.	$J_{n+1}(x) = x\frac{n}{n+1}J_n(x) - J_{n-1}(x), J_1(x) = x^2, J_2(x) = x^3 - 1$
11.	$K_{n+1}(x) = x\frac{n}{n^2+1}K_n(x) - K_{n-1}(x), K_1(x) = x, K_2(x) = x^2 - 1$
12.	$L_{n+1}(x) = xL_n(x) - \frac{n}{n+1}L_{n-1}(x), L_1(x) = 1, L_2(x) = x + 1$