Лекция 3

Подпрограммы.

Разработка алгоритмов и программ сверху вниз

Описание функции (подпрограммы)

Заголовок функции

{

Объявление локальных переменных

Операторы

}

Заголовок функции имеет вид:

тип имя_функции (список_параметров)

Вызов функции

Чтобы выполнить подпрограмму, нужно ее вызвать (обратиться к ней). Вызов осуществляется по имени функции и имеет вид:

имя_функции (аргумент1, аргумент2,…)

Аргументами могут быть ***константы, переменные или выражения.***

Пример программы с подпрограммой

- *Задача.* Даны натуральные числа *n* и *m* (n>m). Определить *c = n! / (m! * (n-m)!)*

Схема функции fact - вычисления k!

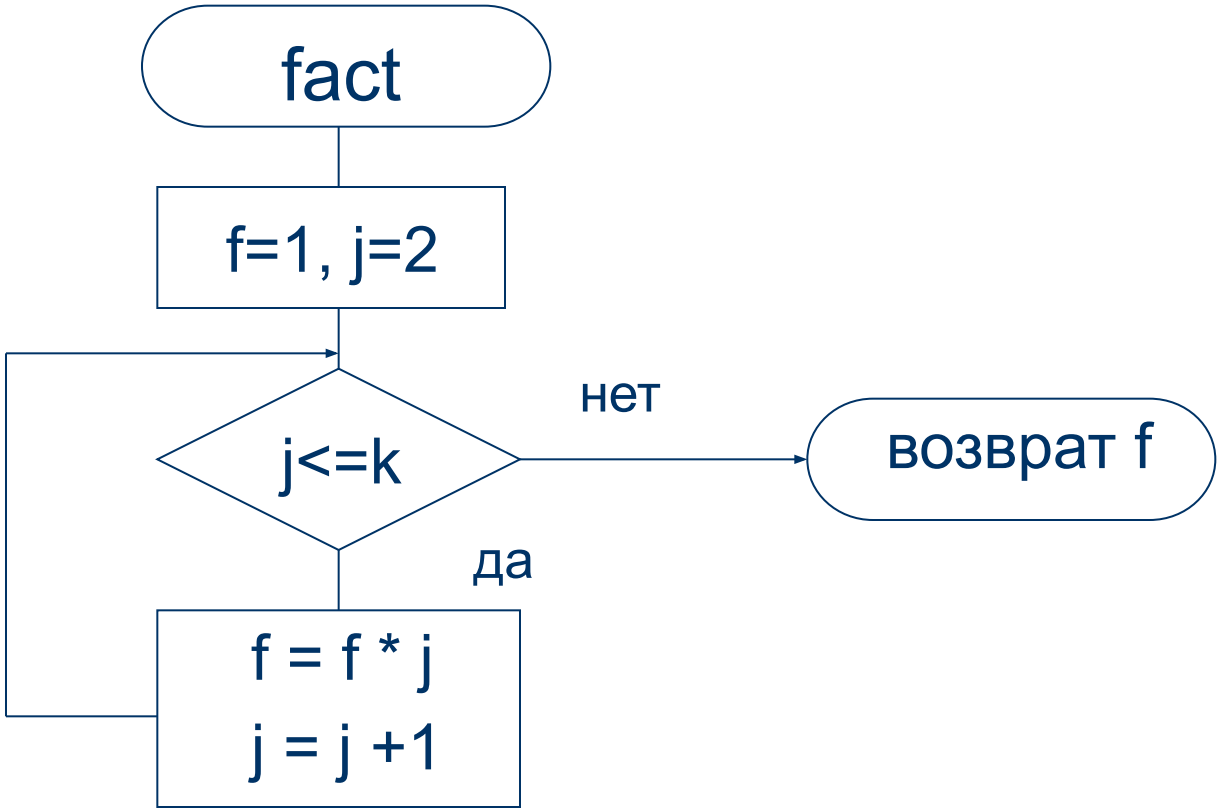
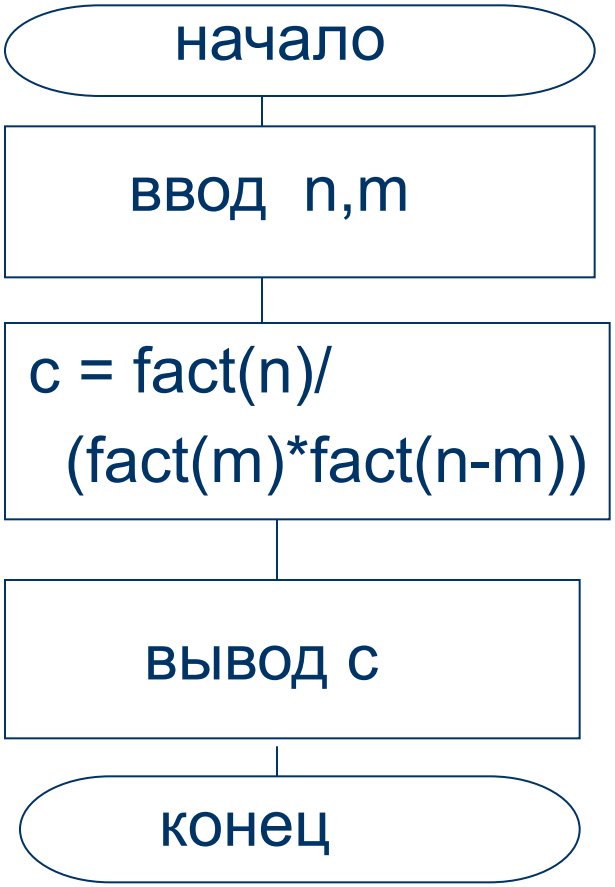


Схема главной функции main



Программа

```
#include <stdio.h>
long fact (int k);   /* прототип функции */
     /*       Главная функция     */
void main()
{  int n, m, c;  /* исходные данные и результат */
   printf("\nВведите два исходных целых числа (n,m):");
   scanf("%d %d", &n, &m);
   c = fact(n) / (fact (m) * fact (n-m));
   printf ("\n c = %d", c);
 }
```

/* Функция вычисления k! */

```
long fact (int k)
{   long f;        // k!
    int j;      // текущий множитель
    f=1;
    for (j=2; j<=k; j++)
       f = f * j;
    return f;       // возврат значения функции
}
```

Разработка алгоритмов и программ сверху вниз

Задача. *Дано целое* ***n*** *и вещественные*

X_1 X_2 ... X_n

Составить программу печати заданных вещественных чисел в порядке возрастания (не убывания).

Вход:

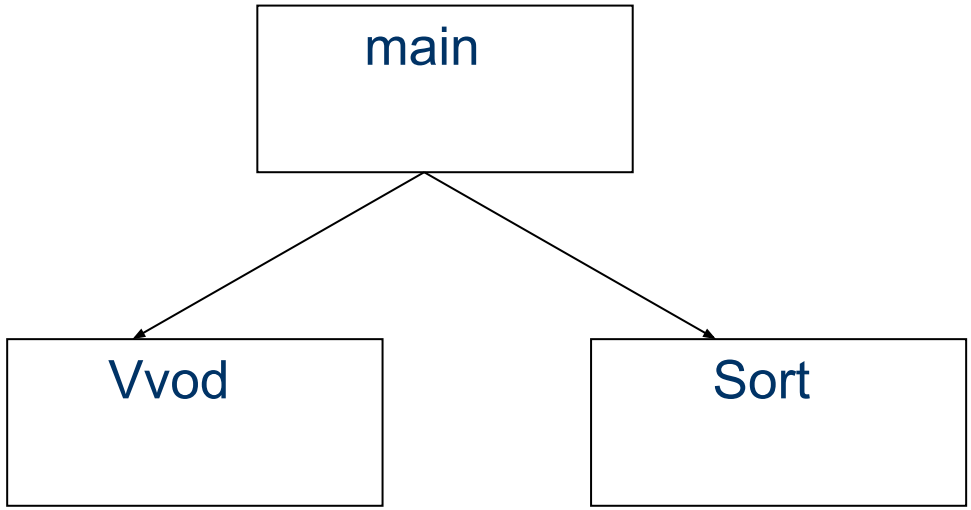
Введите количество чисел: 5

Введите числа: 12.5 6 14 -3 10

Выход:

Упорядоченные числа:-3.0 6.0 10.0 12.5 14.0

Функциональная структура программы



1 этап.

Разработка алгоритма функции *main().*

Алгоритм :

- 1. n = Vvod(x); /* Ввод ***n*** и массива ***x*** */
- 2. Sort (x,n); /* Сортировка массива ***x*** по возрастанию*/
- 3. Вывод сортированного по возрастанию массива ***x***

2 этап

Разработка подпрограмм

Алгоритм функции ввода данных

```
int  Vvod (float x[])
{
    Ввод n;
    for (i=0; i<n; i++)
        Ввод x[i];
    Возврат n;
}
```

Вывод массива x

Вывод заголовка "Упорядоченные числа:";

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

    Вывод x[i];

Метод последовательного нахождения максимума

2.5 6 14 -3 10 // рассматривается n элементов
2.5 6 10 -3 14

2.5 6 10 -3 // рассматривается n-1 элементов
2.5 6 -3 10

2.5 6 -3 // рассматривается n-2 элементов
2.5 -3 6

2.5 - 3 // рассматривается n-3 элементов
-3 2.5

Алгоритм функции сортировки массива *x* по возрастанию

```
void   Sort (float  x[],  int  n)
{   for (k=n-1; k>0; k--)
    { Определение максимума среди
        элементов x[0], ... , x[k]   и   его
        индекса     imax.
      Обмен:  x[imax] <--> x[k];
    }
}
```

3 этап

Определение максимума среди элементов x[0], ... , x[k] и его индекса imax.

Фрагмент программы:

```
imax =0;
for (i =1; i <= k; i++)
        if  (x[i] > x[imax])   imax = i;
```

Программа:

```
#include <stdio.h>
#define  NMAX  100  /* Макс. кол-во чисел*/
     /* Функция ввода данных * /
int  Vvod (float x[])
{
      int   n;       /* Количество чисел     */
      int   i;       /* Индекс текущего числа     */
```

```
    printf ("\nВведите количество чисел\n");
    scanf ("%d", &n);
    printf ("Введите числа\n");
    for (i=0; i<n; ++i)
      scanf("%f", &x[i]);
    return n;
}
```

/* Функция сортировки массива x по возрастанию */

```
void   Sort (float  x[],  int  n)
{
   int   k;        /* Максимальный индекс просмотра*/
   float r;  /* Для обмена                */
   int  imax;   /*  Индекс максимального элемента */
   int   i;        /* Индекс текущего числа          */
```

```
    for (k=n-1; k>0; k--)
    {  imax =0;
       for (i =1; i <= k; i++)
           if  (x[i] > x[imax])   imax = i;
       /*  Обмен   x[imax] и  x[k]  */
       r = x[imax];
       x[imax] = x[k];
       x[k] = r;
    }
}
```

/* Главная функция */

```
void main ()
{   float x[NMAX];    /* Обрабатываемые числа   */
    int   n;          /* Количество чисел       */
    int   i;          /* Индекс текущего числа  */
       /* 1. Ввод массива x  */
    n = Vvod(x);
       /* 2. Сортировка массива x по возрастанию */
    Sort(x,n);
```

```
        /* 3. Вывод массива x */
    printf("Упорядоченные числа:\n");
    for (i=0; i<n; ++i)
        printf ("  %.1f", x[i]);
}
```