ФУНКЦИИ.

– описание
– вызов

1.Описание функции

Заголовок

```
<тип> name_fun(<тип1>арг1,<тип2>арг2,…<тип n>арг n)
```

```
{
описание 1;
. . .
описание m;
```

Описание локальных переменных

```
оператор_1;
. . .
оператор_k;
}
```

Операторная часть

где:

<тип>

int

float

void

name_fun

(<тип1> арг1,<тип2> арг2,…<тип n> арг n)

список формальных аргументов с указанием их типов

ВЫЗОВ ФУНКЦИИ

1. Если функция описана как `void` и не возвращает значений, то ее вызов будет самостоятельным оператором и имеет вид:

```
name_fun ( aprf1, aprf2,… aprfn)
```

где:

`aprf1, aprf2,… aprfn` список фактических аргументов

Пример:
составить функцию, которая определяет
большее значение из двух заданных чисел

```
f  = max(x,y)
```

и печатает результат в самой функции.

```
void  f_max (float a, float b)
{
    float max;
    if(a>b)   max = a;
          else max = b;
    printf ("\n %4.1f", max);
}
int main()
 { float  x, y, z;
   scanf("%f%f",&x,&y);
   f_max(y, x);
 }
```

2. Если в описании функции указан тип,
то вызов функции не является самостоятельным оператором,
а должен записываться внутри какого-либо другого оператора,
например, оператора присваивания, условного оператора и т.п.

```
b=name_fun ( aprf1, aprf2,… aprfn)
```

Пример:
составить функцию, которая определяет большее значение из двух заданных `f = max(x, y)` и распечатать результат в основной программе.

```
float fmax (float a, float b)
{
    float max;
     if(a>b)   max = a;
     else max = b;
    return max;
}

void main()
{
float  x, y, z;
scanf("%f%f",&x,&y);
z =f max(y, x);
printf("\n%4.1f", z);
}
```

Месторасположение описания функций

1) Описание функции до основной программы

```
/*описание функции*/
void  myfun ( int x, int y)
{
. . .
}
/*основная программа*/
int main()
{. . .
/*вызов функции*/
myfun (x, y);
. . .
}
```

2)Описание функции идет после основной программы

/*объявление прототипа функции (указание заголовка)*/

```
void myfun (int x, int y);
```

/*основная программа*/

```
int main()
{. . .
```

/*вызов функции*/

```
myfun (x, y);
}
```

/*описание функции*/

```
void  myfun ( int x, int y)
{
. . .
}
```

ВАРИАНТЫ ОПИСАНИЯ И ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ.

1. Функция не возвращает значений.

a) при описании должен быть указан функции **`void`**;

b) вызов функции является самостоятельным оператором.

а)Функция не имеет аргументов

Пример1: Рассчитать y = x*sin(x) для 6 значений, начиная с x=0 с шагом dx=0.1. Результаты распечатать в самой функции.

```
void prim ()
{
int i, n=6;
float x=0, y;
for (i=1; i<=n; i++)
{
y=x*sin(x);
printf  (“%f %f\n”,  x, y);
x+=0.1;
}
}
void main ()
{
prim();
}
```

б)Функция имеет аргументы

Пример2: Рассчитать y = x*sin(x) для n значений,
начиная с x=0 с шагом dx=0.1.Результаты распечатать в самой функции.
В качестве передаваемого аргумента используем количество вычислений n.

```
void prim (int n)
{ int i;
float x=0, y;
for(i=1; i<=n;  i++)
{
y=x*sin(x);
printf("%f_%f\n",x, y);
x=x+0.1;
}
}
```

```
void main()
{
int n=6;
prim (n);
}
```

2.Функция возвращает одно значение через свое имя.

a) при описании должен быть указан конкретный тип функции;

b) внутри функции должна быть описана дополнительная переменная,

c) тип который совпадает с типом функции и которая по сути хранит

d) результаты вычислений;

e) внутри функции должен присутствовать оператор `return`;

f) вызов функции не является самостоятельным оператором.

Пример:
составить функцию, которая определяет большее значение из трех заданных чисел f = max(x, y,z) и распечатать результат в основной программе.

```
float fmax (float a, float b, float c)
{
    float max;
     if(a>b)    max=a;
                else max=b;
     if(c>max)    max=c;
    return max;
}
int main()
{
float  x, y, z, rez;
scanf("%f%f%f", &x,&y,&z);
rez=fmax(y,x,z);
printf("\n rez =%4.1f", rez);
}
```