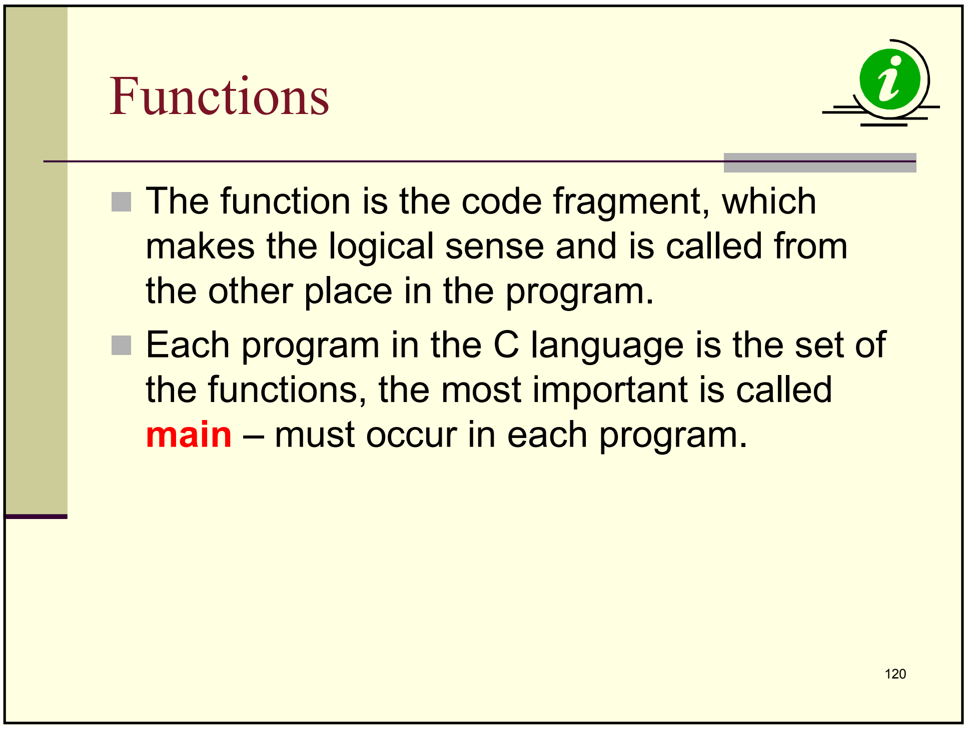
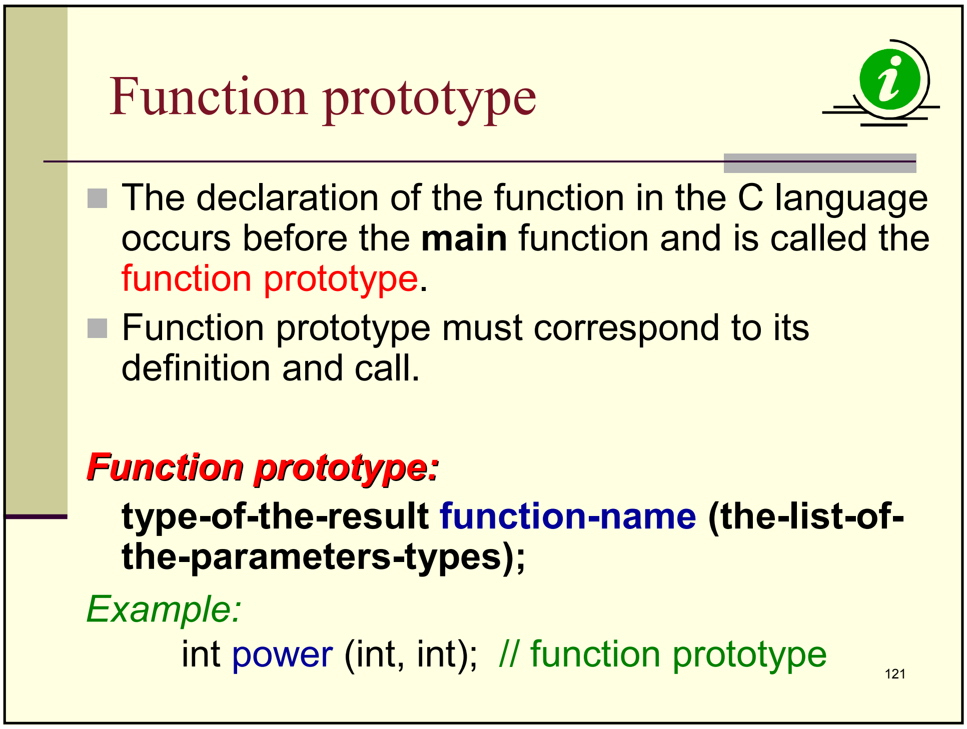


Functions

- The function is the code fragment, which makes the logical sense and is called from the other place in the program.
- Each program in the C language is the set of the functions, the most important is called **main** – must occur in each program.



Function prototype

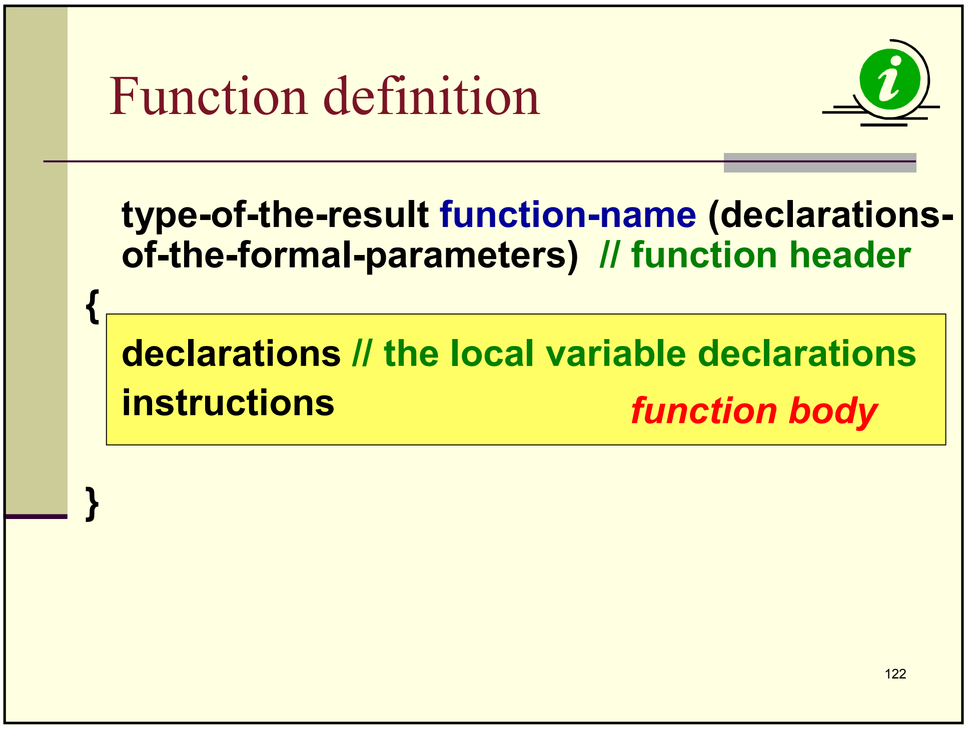
- The declaration of the function in the C language occurs before the **main** function and is called the function prototype.
- Function prototype must correspond to its definition and call.

Function prototype:

type-of-the-result function-name (the-list-of-the-parameters-types);

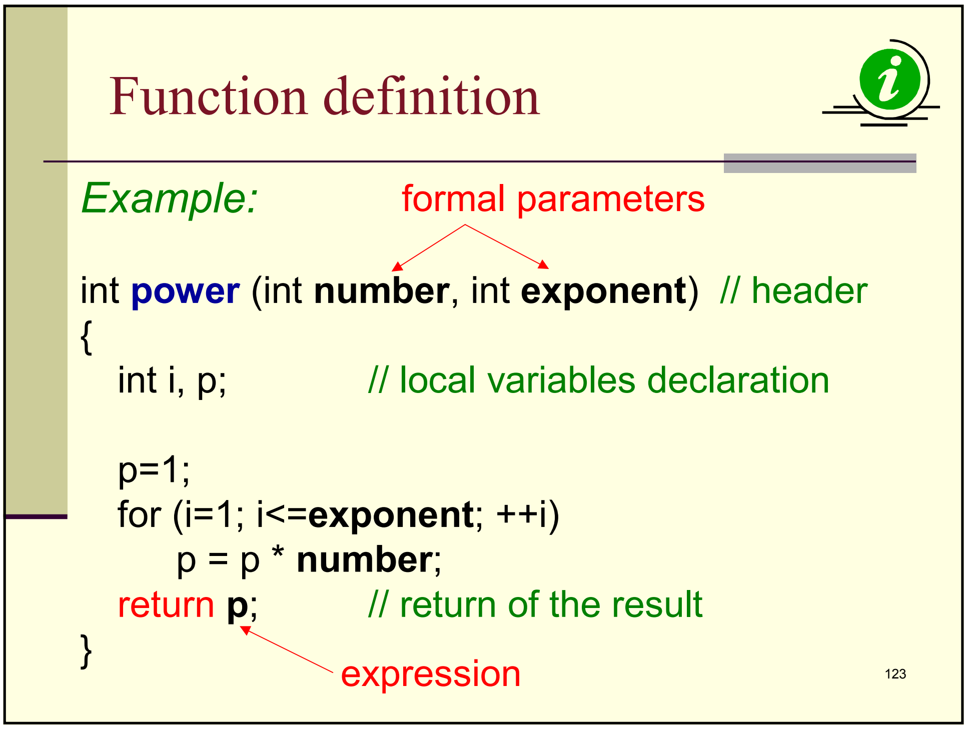
Example:

```
int power (int, int);  // function prototype
```



Function definition

```
    type-of-the-result function-name (declarations-
    of-the-formal-parameters)  // function header
{
    declarations // the local variable declarations
    instructions
                                        function body
}
```



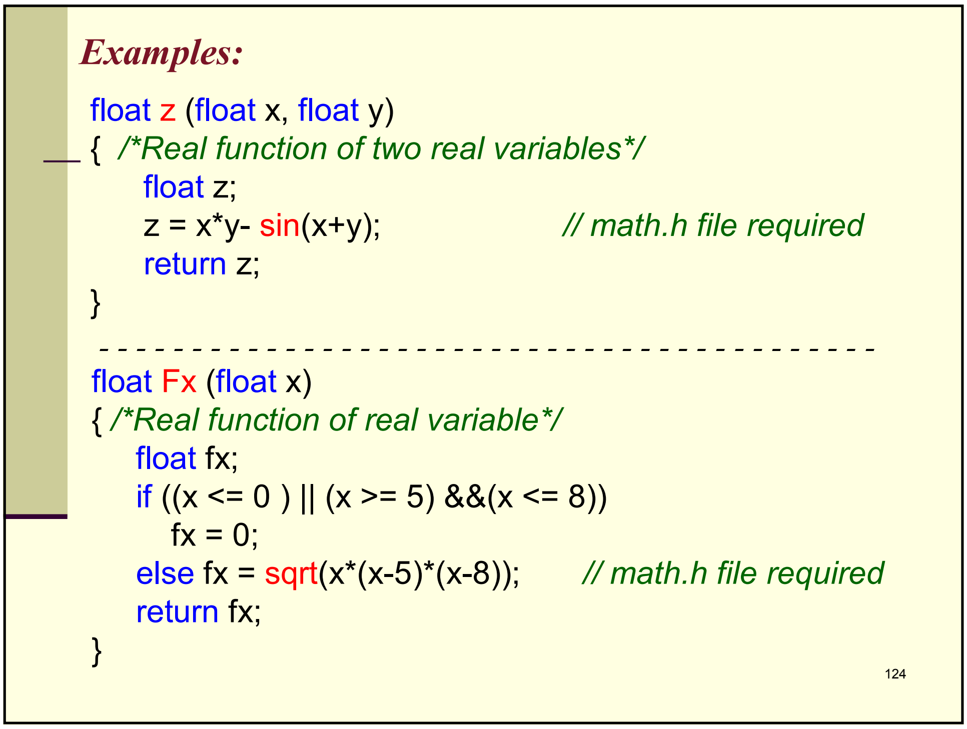
Function definition

Example:

formal parameters

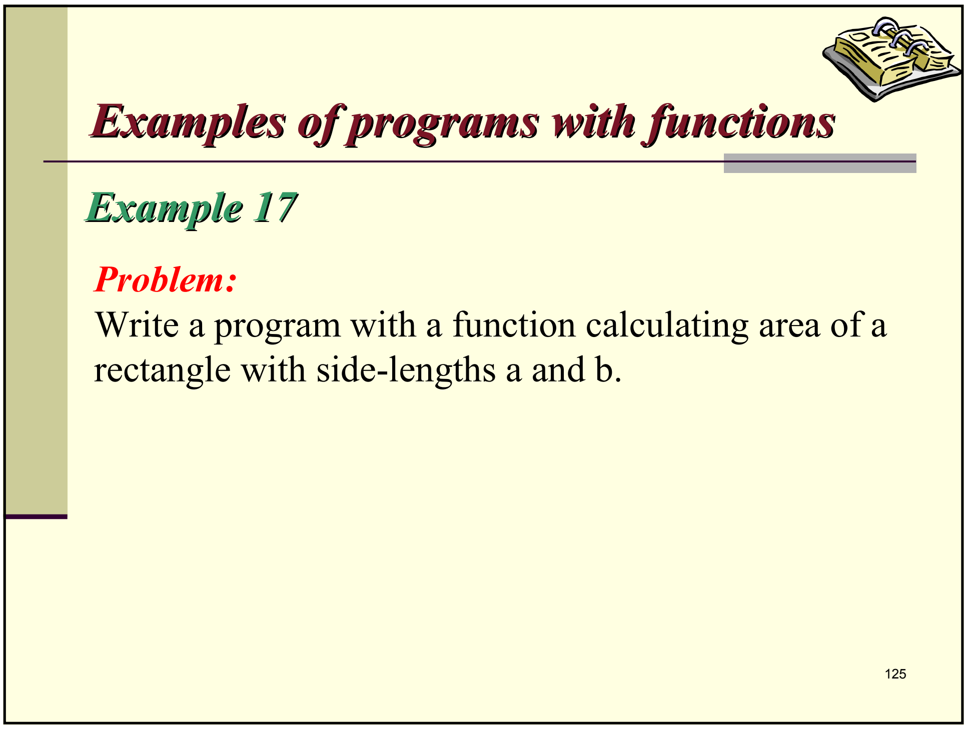
```
int power (int number, int exponent)  // header
{
   int i, p;           // local variables declaration

   p=1;
   for (i=1; i<=exponent; ++i)
        p = p * number;
   return p;           // return of the result
}
```

expression



Examples:

```
float z (float x, float y)
{  /*Real function of two real variables*/
      float z;
      z = x*y- sin(x+y);                  // math.h file required
      return z;
}
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
float Fx (float x)
{ /*Real function of real variable*/
     float fx;
     if ((x <= 0 ) || (x >= 5) &&(x <= 8))
        fx = 0;
     else fx = sqrt(x*(x-5)*(x-8));       // math.h file required
     return fx;
}
```

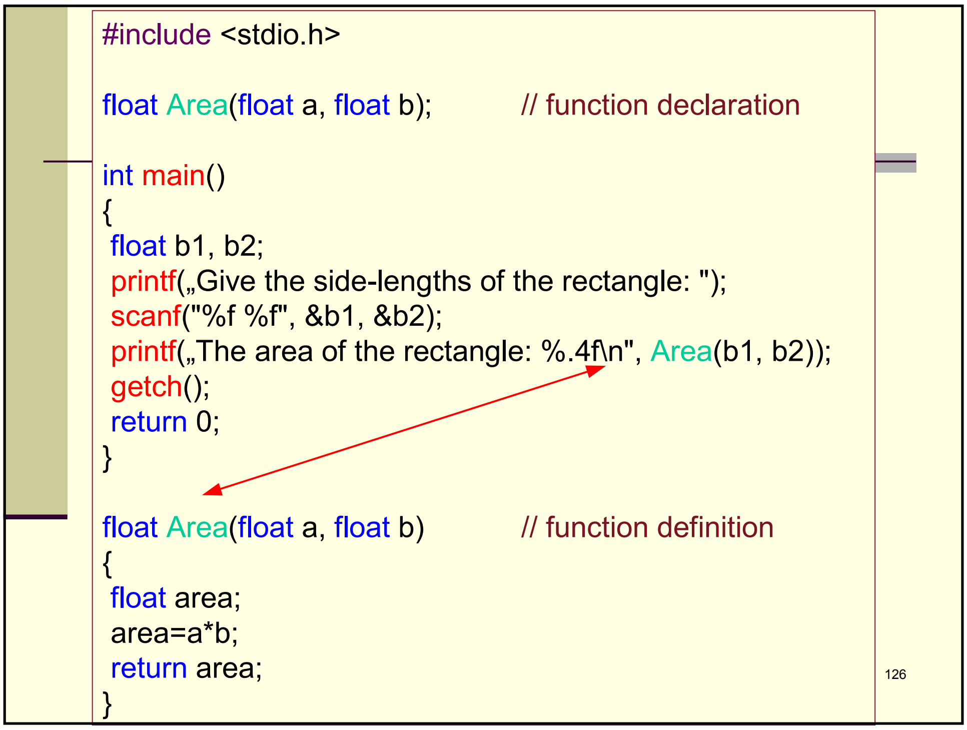


Examples of programs with functions

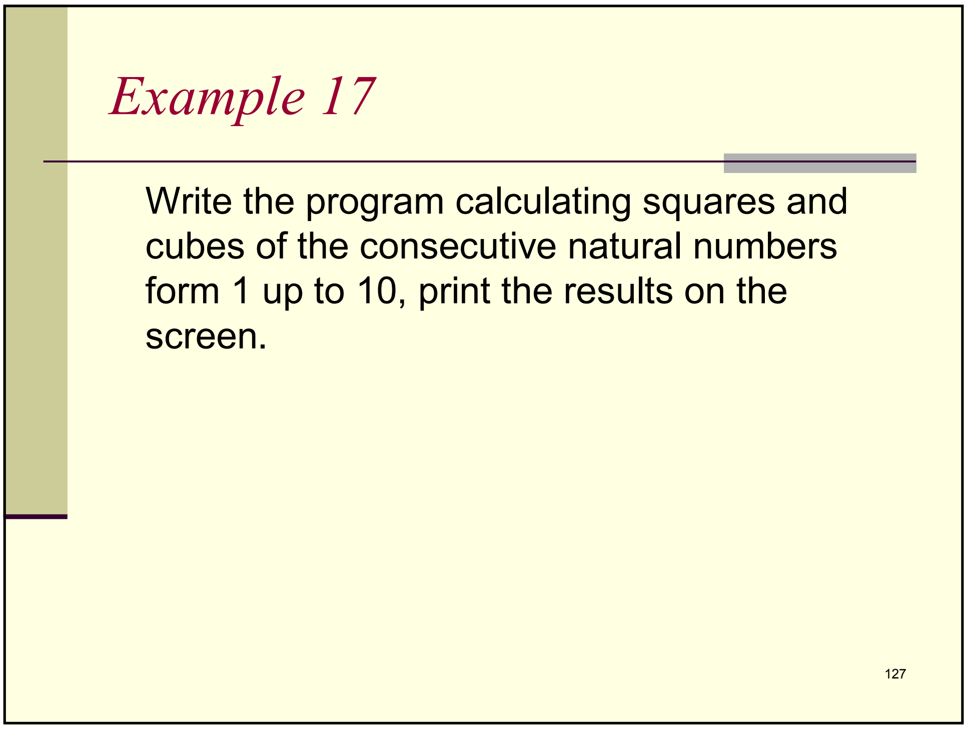
Example 17

Problem:

Write a program with a function calculating area of a rectangle with side-lengths a and b.

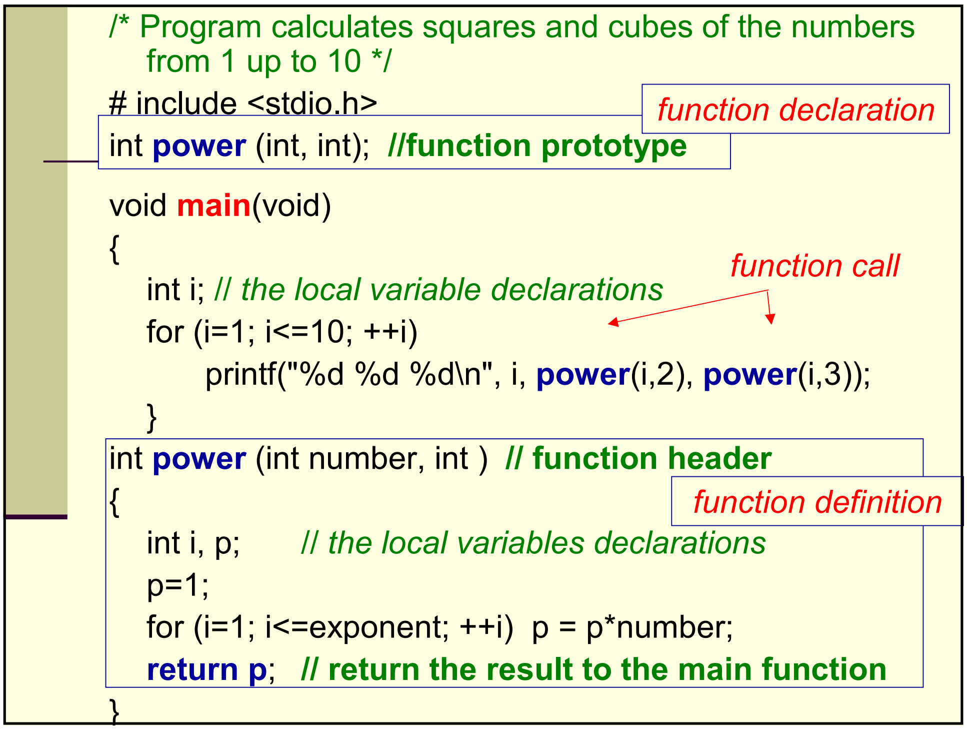


```
#include <stdio.h>

float Area(float a, float b);          // function declaration

int main()
{
 float b1, b2;
 printf(„Give the side-lengths of the rectangle: ");
 scanf("%f %f", &b1, &b2);
 printf(„The area of the rectangle: %.4f\n", Area(b1, b2));
 getch();
 return 0;
}

float Area(float a, float b)           // function definition
{
 float area;
 area=a*b;
 return area;
}
```



Example 17

Write the program calculating squares and cubes of the consecutive natural numbers form 1 up to 10, print the results on the screen.



```
/* Program calculates squares and cubes of the numbers
   from 1 up to 10 */
# include <stdio.h>
int power (int, int);  //function prototype

void main(void)
{
   int i; // the local variable declarations
   for (i=1; i<=10; ++i)
        printf("%d %d %d\n", i, power(i,2), power(i,3));
   }
int power (int number, int )  // function header
{
   int i, p;      // the local variables declarations
   p=1;
   for (i=1; i<=exponent; ++i)  p = p*number;
   return p;   // return the result to the main function
}
```

function declaration

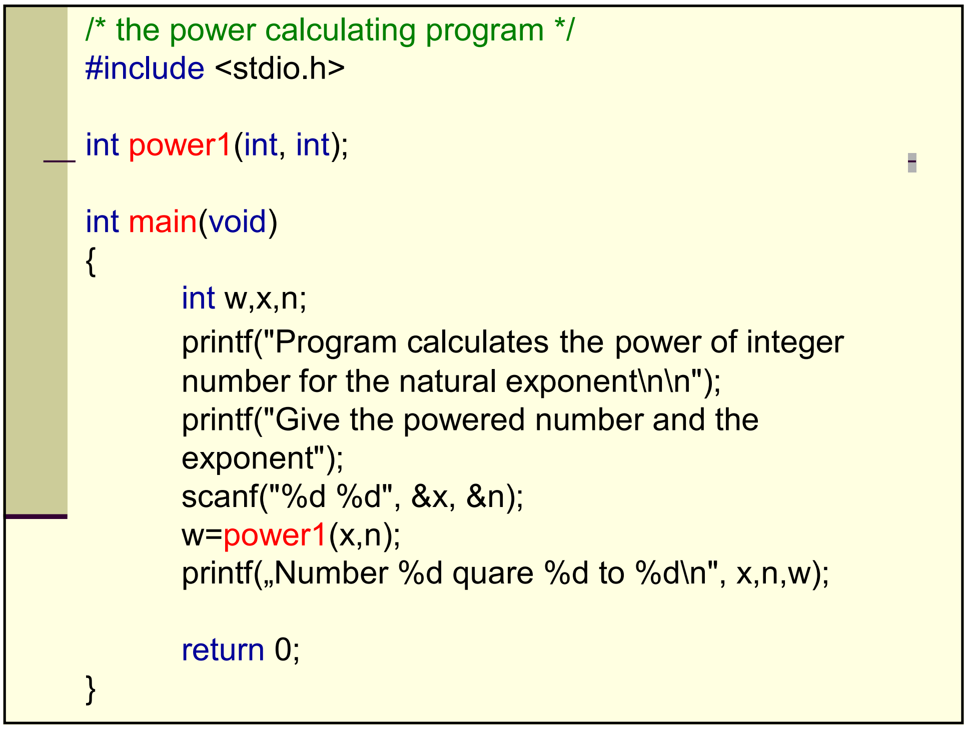
function call

function definition

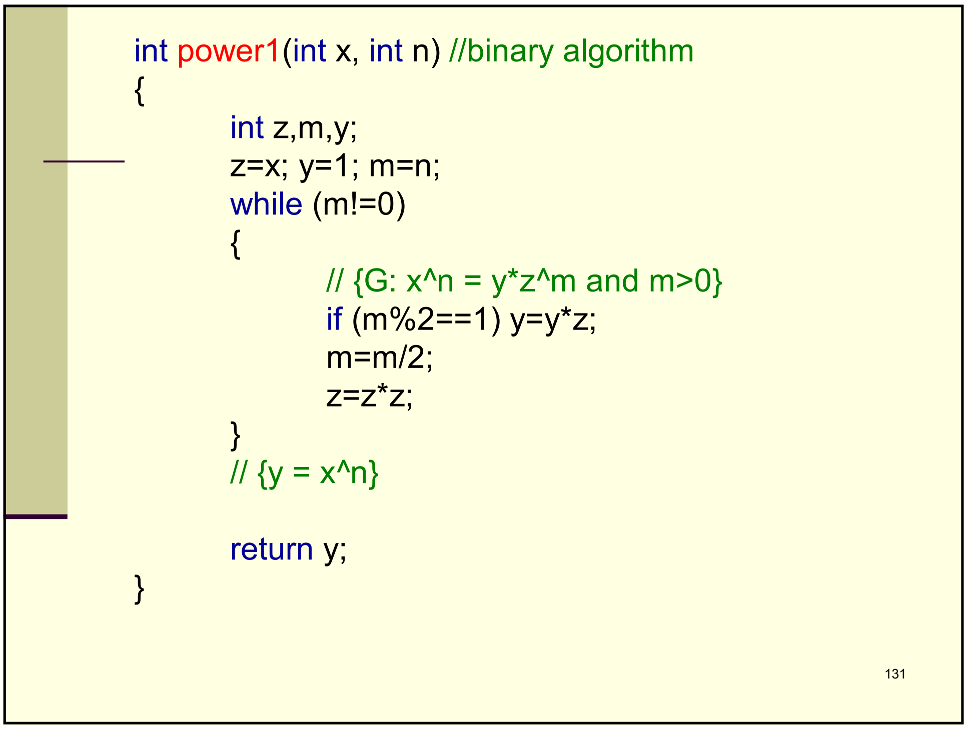
Example 18

Write the program calculating powers for any natural number and natural exponent and printing the results on the screen.

Use the binary algorithm of power calculating.



```
/* the power calculating program */
#include <stdio.h>

int power1(int, int);

int main(void)
{
        int w,x,n;
        printf("Program calculates the power of integer
        number for the natural exponent\n\n");
        printf("Give the powered number and the
        exponent");
        scanf("%d %d", &x, &n);
        w=power1(x,n);
        printf(„Number %d quare %d to %d\n", x,n,w);

        return 0;
}
```



```
int power1(int x, int n) //binary algorithm
{
        int z,m,y;
        z=x; y=1; m=n;
        while (m!=0)
        {
                // {G: x^n = y*z^m and m>0}
                if (m%2==1) y=y*z;
                m=m/2;
                z=z*z;
        }
        // {y = x^n}

        return y;
}
```