



函数



中北大学 ACM-ICPC 创新实验室

115.28.80.81

ACM实验室课堂问答系统V1.0

姓名

输入姓名

学号

输入学号

问题描述

提交

ACM实验室课堂问答系统V1.0

提问者学号: 1514010** 提问时间: 2016-10-28 21:01:34

PPT演示专用语句

提问者学号: 1514010** 提问时间: 2016-10-28 21:00:23

PPT演示专用语句!!!

← 上一页

Design By BBBBaobao

没有加群的可以
点击这里加群

$y=f(x)$ 数学函数

• 输入

x

• function

f()

• 用来进行运算 $\sin x + e^x @ ! \# \$ \sim$

• 输出

y

计算机函数

计算机函数

编辑

计算机的函数，是一个固定的一个程序段，或称其为一个子程序，它在可以实现固定运算功能的同时，还带有一个入口和一个出口，所谓的入口，就是函数所带的各个参数，我们可以通过这个入口，把函数的参数值代入子程序供计算机处理；所谓出口，就是指函数的函数值，在计算机求得之后，由此口带回给调用它的程序。

中文名	计算机函数	应用学科	计算机科学、数学等
外文名	Computer Function	适用领域范围	数学、计算机编程等
别 称	程序函数	分 类	计算机逻辑基础

函数 [编辑]

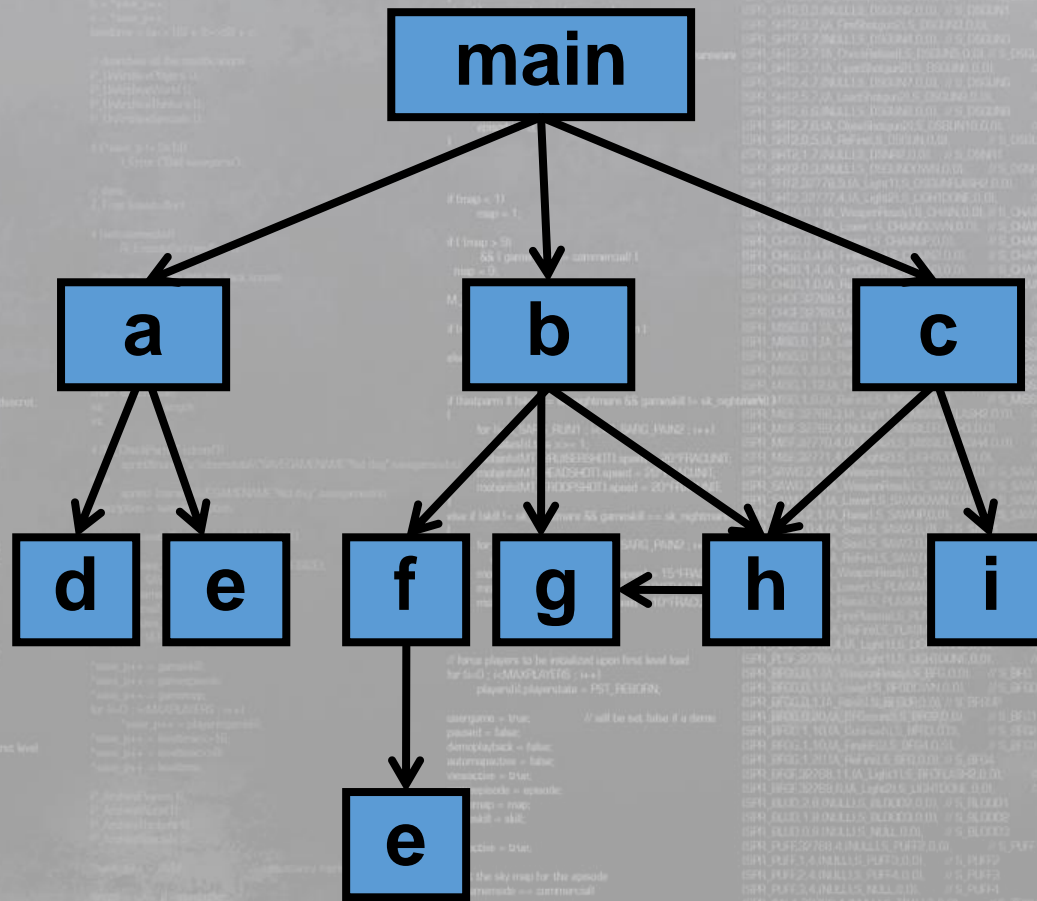
函数（function），又称函数，是一种子程序，利用函数名称，可以接收回传值。例如：

$c = \max(a, b);$

计算机函数

- C 程序可由一个主函数和若干个其他函数构成
- 主函数调用其他函数，其他函数也可以互相调用
- 同一个函数可以被一个或多个函数调用任意多次

C语言程序的一般结构



当一个程序想干的活大小不同时 整个程序写起来就不同

```
#include<stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

```
void see()
{

}

void sw(int x1,int y1,int

)
{

}

int FF = 0;
int ctrl()
{

}

void ini()
{

}

void forWin()
{

}

int main()
{
    ini();
    minn = bfs();
    see();

    while(!ctrl());
    see();

    forWin();
    system("pause");
    return 0;
}
```


模块化程序设计

- **基本思想：**将一个大的程序按功能分割成一些小模块。每个小模块根据需要还可以继续细分,直到分解成一个个功能独立的模块为止。
- **特点：**
 - **各模块相对独立、功能单一、结构清晰、接口简单,容易理解**
 - **各模块可以独立设计算法,单独编写和测试**
 - **一个模块中的错误不易扩散和蔓延到其它模块**
 - **众人可同时进行集体性开发**
 - **易于维护和功能扩充**

函数的使用

声明

定义（一般和
上一步在一起）

调用

函数要先申明再调用

```
#include<stdio.h>
int fun(int x); // 声明
int fun(int x) // 定义
{
    return x+1;
}

int main()
{
    fun(); // 调用
    return 0;
}
```

```
#include<stdio.h>
int fun(int x) // 声明+定义
{
    return x+1;
}
```

函数的定义

定义有参函数的一般形式为：

类型名 函数名（形式参数类型 参数名）

函数体

多个参数用逗号隔开

```
int fun(int x)
{
    return x+1;
}
```

```
int fun(int x,int y)//声明+定义
{
    return x+y;
}
```

函数的返回值

返回语句 (return)

函数返回值则需要调用return语句来完成。该语句的一般形式：

- return(表达式)；或 return 表达式；
- 例如： return;
- return z;
- return (z);
- return (x>y?x:y);

• 函数返回值的类型即函数值的类型，函数值的类型以函数定义的类型为准

```
int fun(int x)
{
    return x+1;
}
```

The diagram shows a C++ function definition. The return type 'int' is in black, and the function name 'fun' is in red. The parameter '(int x)' is in black. The opening curly brace '{' is in black. The return statement 'return x+1;' is on a light blue background. 'return' is in black, 'x' is in black, and '+1;' is in red. The closing curly brace '}' is in black. A blue arrow points from the 'return' keyword to the 'int' return type.

无参函数

无返回值函数

类型名 函数名 (形式参数表列)

函数体

```
#include<stdio>
```

```
int lalla()// 申明+定义  
{
```

```
    printf("123\n");  
    return 1;  
}
```

```
int main()
```

```
{  
    lalla()// 调用  
}
```

```
void lalla(int a)// 申明+定义  
{
```

```
    printf("123\n");  
    return ;  
}
```

```
int main()
```

```
{  
    lalla(a)// 调用  
}
```

啥活也能干

```
#include<stdio>
```

```
void lalla()//申明+定义
```

```
{
    printf("123\n");
    return ;
}
```

```
int main()
```

```
{
    lalla();//调用
}
```

```
#include<stdio>
```

```
int main()
```

```
{
    printf("123\n");
}
```

函数调用时的参数传递

形式参数和实际参数

◆在调用有参函数时，主调函数和被调函数之间有数据传递关系

◆定义函数时函数名后面的变量名称为“形式参数”（简称“形参”）

◆主调函数中调用一个函数时，函数名后面参数称为“实际参数”（简称“实参”）

◆实际参数可以是常量、变量或表达式



函数的调用

类型名 函数名 (形式参数表列)

函数体

当作一个变量使用
同 a 和 $1+1$ 一样

```
#include<stdio.h>
int fun(int x) // 声明+定义
{
```

```
    return x+1;
}
```

```
int main()
{
```

```
    int a = 2;
```

```
    printf("%d\n", fun(1)); // 调用
```

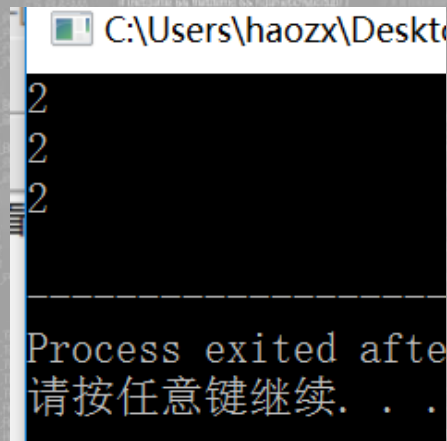
```
    printf("%d\n", 1+1);
```

```
    printf("%d\n", a);
```

```
    return 0;
```

```
}
```

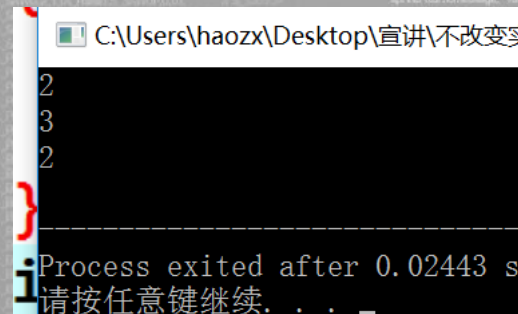
实际参数



函数调用时不改变实参的值

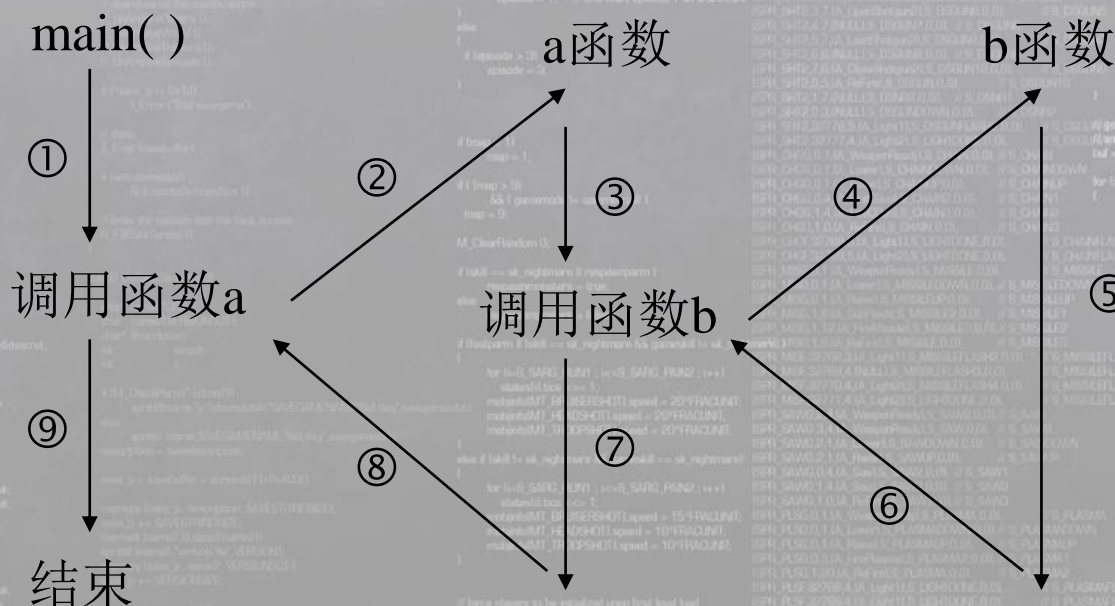
```
#include<stdio.h>
int fun(int x)//声明+定义
{
    x = x + 1;
    return x;
}
int main()
{
    int a = 2;
    printf("%d\n",a);
    printf("%d\n",fun(a));
    printf("%d\n",a);
    return 0;
}
```

给到fun只是a的值而不是吧a给过去



函数的嵌套调用

即调用一个函数的过程中，又可以调用另一个函数



eg.

<http://acm.hdu.edu.cn/showproblem.php?pid=2028>

Lowest Common Multiple Plus

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 52848 Accepted Submission(s): 21911

Problem Description

求n个数的最小公倍数。

Input

输入包含多个测试实例，每个测试实例的开始是一个正整数n，然后是n个正整数。

Output

为每组测试数据输出它们的最小公倍数，每个测试实例的输出占一行。你可以假设最后的输出是一个32位的整数。

Sample Input

```
2 4 6
3 2 5 7
```

Sample Output

```
12
70
```

局部变量和全局变量

- 在函数的开头定义。
(局部)
- 在函数内的复合语句定义。(局部)
- 在函数的外部定义。
(全局)
- 局部变量的优先级大于全局的变量。

```
#include<stdio.h>

int a = 1;
int b = 1;
int main()
{
    int a = 2;
    int c = 3;
    printf("%d\n", a);
    printf("%d\n", b);
    printf("%d\n", c);
    return 0;
}
```

在函数内通过全局变量改变主函数内变量

```
#include<stdio.h>

int b;

int change(int x)
{
    return x+1;
}

void change2()
{
    b = b + 1;
    return;
}

int main()
{
    int a = 1;
    b = 10;
    a = change(a);
    change2();
    printf("%d\n",a);
    printf("%d\n",b);
    return 0;
}
```

C:\Users\haozx\Desktop\宣

2
11

Process exited after 0.
请按任意键继续. . .

头文件

► #include <>

- 这里的include 称为文件包含命令，其意义是把尖括号<>或引号” ” 内指定的文件包含到本程序来，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h。因此也称为头文件或首部文件。C语言的头文件中包括了各个标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。
- 有关字符串处理的函数都在 #include <string.h>
- 有关字符函数都在 #include <ctype.h>
- 有关数学的函数都在 #include <math.h>
- 有关输入输出的都在 #include <stdio.h>

自己定义函数的优先级 比头文件优先级大

