中国人民大学
RENMIN UNIVERSITY OF CHINA

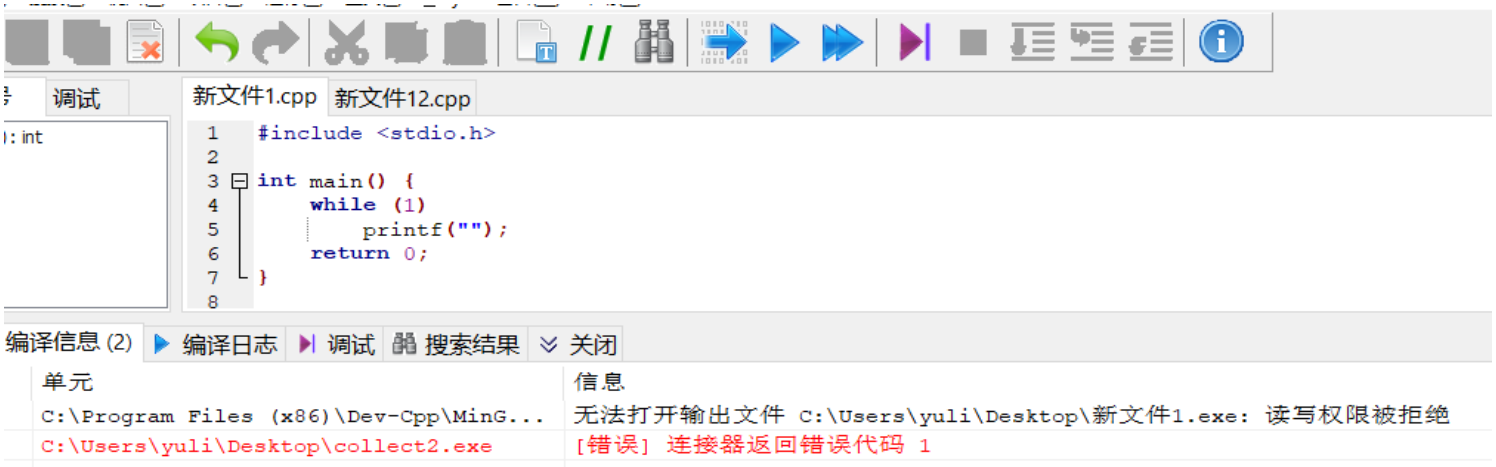
C程序设计

第7讲 函数

余力

buaayuli@ruc.edu.cn

编译读写权限被拒绝



遇到这种情况，多般情况就是这个.cpp 文件被编译运行次数多，通常是因为一个程序还没有运行完（比如死循环，但你可能不知道），然后你又按了“编译运行”，这种情况下经常就会出现这种情况。如果出现这种情况，可能关闭 Dev-c++也还是不行，除非重启电脑。

解决方法是：

第 1、为避免这种情况，大家一般不要同时编译运行多个程序，如果一个程序没有运行完，你又编译运行另一个，很可能就会出现这种情况。大家平时尽可能不要同时打开和运行多个题目的程序，执行完一个后就关闭一个，再开始一个新的。

第 2、如果出现上述情况，一个应急的处理办法是，新建一个文件，将之前的代码全部复制到该新文件中，重新编译运行。

不正常运行情况

```c
#include <stdio.h>

int main() {
    printf("Good");
    return 100;
}
```

```
Good
--------------------------------
Process exited after 0.2208 seconds with return value 100
请按任意键继续. . .
```

3221225477: 访问越界，一般是读或写了野指针指向的内存。

3221225725: 堆栈溢出，数组开得太大，无穷调用递归。

3221225620: 除 0 错误，一般发生在整型数据除了 0 的时候。

```
--------------------------------
Process exited after 0.3552 seconds with return value 3221225725
请按任意键继续. . .
```

内容提要

1 函数的基本用法

2 数组名作为函数参数

3 全局变量与局部变量

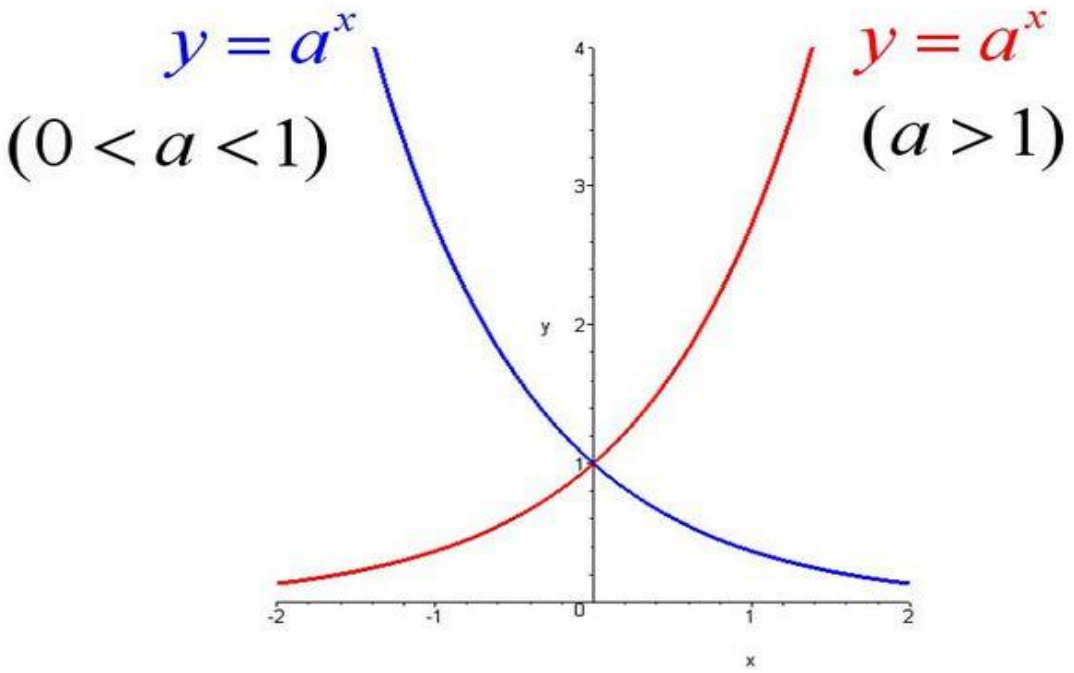
4 函数嵌套调用

5 函数应用实例

中国人民大学
RENMIN UNIVERSITY OF CHINA

1. 函数的基本使用

你熟悉的函数



y=sin(x)

y=3x+5

$y=x_1+x_2$

$y=\max(x_1, x_2)$

y=f(x) 一个函数表示一个输入与输出的对应关系

一个函数就是独立负责一件事！

函数是一种分工模式，模块化编程！

#492 加法表达式

```
long long int f(int m, int n) {
    long long int i, j, sum = 0;
    for (i = m; i < n; i++)
        sum = sum + pow(10, n - i - 1) * (s[i] - 48);
    return (sum);
}
```

```
for (i = 1; i <= strlen(s); i++)
    for (j = i + 1; j <= strlen(s); j++)
        for (t = j + 1; t < strlen(s); t++)
            c[e++] = f(0, i) + f(i, j) + f(j, t) + f(t, strlen(s));
```

函数风格

普通风格

```
for (i = 1; i <= length - 3; i++)
    for (j = 1; j <= length - 3; j++)
        for (k = 1; k <= length - 3; k++)
            for (l = 1; l <= length - 3; l++)
                if (i + j + k + l == length) {
                    long long num1 = 0, num2 = 0, num3 = 0, num4 = 0;
                    for (ic = 1; ic <= i; ic++)
                        num1 = num1 * 10; + replace[ic - 1];
                    for (jc = 1; jc <= j; jc++)
                        num2 = num2 * 10 + replace[i + jc - 1];
                    for (kc = 1; kc <= k; kc++)
                        num3 = num3 * 10 + replace[i + j + kc - 1];
                    for (lc = 1; lc <= l; lc++)
                        num4 = num4 * 10 + replace[i + j + k + lc - 1];
                    sum[count] = num1 + num2 + num3 + num4;
                    count++; }
```

493 回文素数数位和

```
int su(int x){//判断素数 1-yes 0-no
    for(int i=2;i*i<=x;i++)
        if(x%i==0)
            return 0;
    return 1;
}
```

```
int hui(int x){//判断回文 1-yes 0-no
    char a[20];
    int num=0;
    while(x!=0){
        a[num++]=x%10;
        x/=10;}
    for(int i=1;2*i<=num;i++)
        if(a[i]!=a[num--i+1])
            return 0;
    return 1;
}
```

```
int digitsum(int x){
    int sum=0;
    while(x!=0){
        sum+=x%10;
        x/=10;}
    return sum;
}
```

```
int main(){
    cin>>m>>n;
    for(int i=m;i<=n;i++)
        if(hui(i)&&su(i)){
            int t=digitsum(i);
            if(t>summax){
                summax=t;
                end=i;
            }
        }
    cout<<end<<"-"<<summax;
    return 0;
}
```

函数的两件事：定义与调用

类型名 函数名()

{

函数体

}

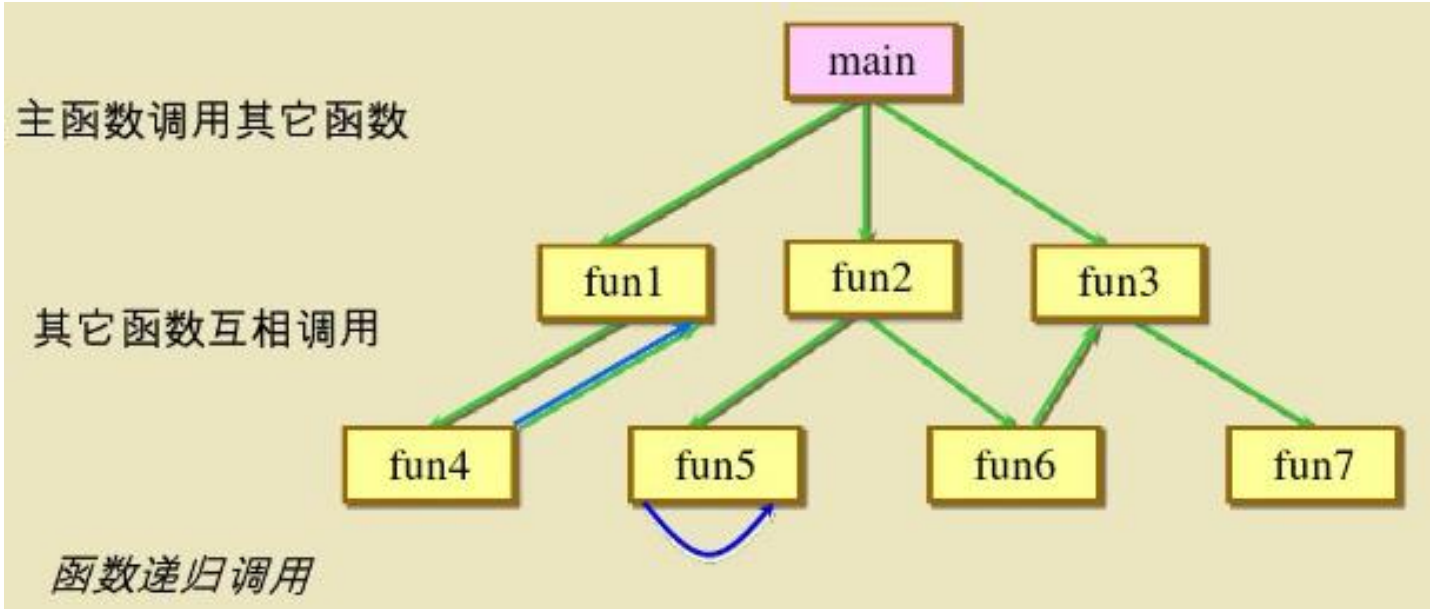
类型名 函数名(void)

{

函数体

}

包括声明部分和语句部分



形式参数（形参）与实际参数（实参）

```
#include <stdio.h>
#include <math.h>
int main()
{   int a;
    printf("请输入整数 a\n");
    scanf( "%d", &a);
    if ( prime(a)==1 )
        printf("%d是质数\n", a);
    else
        printf("%d不是质数\n", a);
}
```

实参

```
int prime(int x )
{
    int i;
    for(i=2; i<= sqrt(x); i++)
        if (x%i==0) break;
    return i>sqrt(x)?1:0;
}
```

形参

+Chp07_判断素数.cpp

参数值的传递

```
#include <stdio.h>
int main()
{ int max(int x, int y);
  int a, b, c;
   printf("two integer numbers:");
   scanf("%d%d", &a,&b);
   c=max(a, b);
printf("max is %d\n", c);
}
```

```
int max(int x, int y)
{
   return(x>y?x:y);
}
```

实参可以是常量、变量或表达式

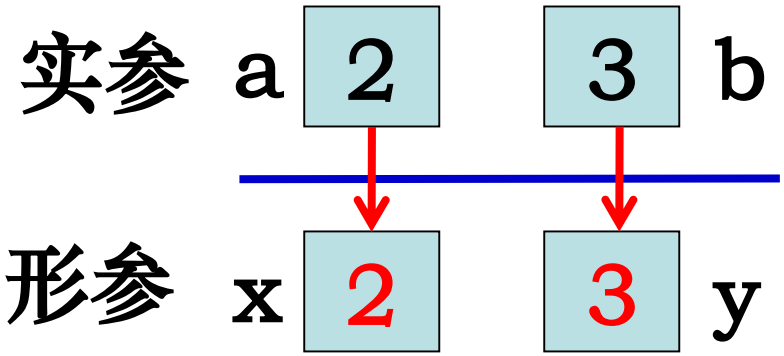
```
two integer numbers:12,-34
max is 12
```

+Chp07_求最大值.cpp

正确理解调用全过程

```
c=max(a,b);          (main函数)
```

```
int max(int x, int y)    (max函数)
{
    int z;
    z=x>y?x:y;
    return(z);
}
```



形参与实参 总结

- 形参：
  - 定义函数时放在函数名后括号中的参数；
  - 未被调用不占内存单元；
  - 被调用后系统为其分配内存单元；
  - 调用结束释放内存单元；
  - 作用域限定在子函数内，属于局部变量
- 实参：
  - 具有确定值的表达式
  - 函数调用时将实在参数赋值给形参变量

思考

```
#include<stdio.h>
void swap1 （int x1, int x2);
int main () {
    int a[2];
    a[0] = 10；  a[1] = 100;
    printf ("%d, %d\n", a[0], a[1]);
    swap (a[0],a[1]);
    printf ("%d, %d\n", a[0], a[1]);
    return 0;
}
    void swap1 (int x1, int x2) {
      int tmp = x1;  x1 = x2;  x2 = tmp;  }
```

什么运行结果？

交换地址变量的值

swap2（&a[0], &a[1]);

```
void swap2 (int* x1, int* x2) {
  int tmp = *x1;
      *x1 = *x2;
      *x2 = tmp;      }
```

交换两个地址所在变量

+Chp07_交换变量值.cpp

函数的返回值

- 通过函数调用使主调函数能得到一个确定的值，这就是函数值
- (1) 函数返回值是通过return语句获得
  - 一个函数中可以有一个以上的return语句，执行到哪一个return语句，哪一个就起作用
  - return语句后面的括号可以不要
- (2) 函数值的类型。
  - 应当在定义函数时指定函数值的类型

```
#include <stdio.h>
int main()
 { int max(float x,float y);
    float a,b;  int c;
    scanf("%f,%f,",&a,&b);
    c= max(a,b);
    printf("max is %d\n",c);
    return 0;
 }
int max(float x, float y)
 {  float z;
    z=x>y?x:y;
    return( z ) ;
}
```

1.5 2.6

2.6 → 2

```
1.5,2.6
max is 2
```

将在max函数中定义的变量z改为float型

+Chp07_求最大值.cpp

还有一件事：函数的声明

- **函数声明：**说明你要用到哪个函数
- 函数调用要具备如下条件：

  (1) 被调函数是已经定义的函数（库函数 or 自己定义的函数）

  (2) 如果使用库函数，写#include

  (3) 如果使用自定义的函数，被调函数需要被声明，或放前面

```
#include <stdio.h>
int main()
{ float add(float x, float y);
   float a,b,c;
   printf("Please enter a and b:");
   scanf("%f,%f",&a,&b);
   c=add(a,b);
   printf("sum is %f\n",c);
   return 0;
}
```

函数声明

函数定义

```
float add(float x, float y)
{ float z;
   z=x+y;
   return(z);
}
```

+Chp07_加法函数.cpp

2. 数组名作参数

数组名作为参数

```
#include<stdio.h>
void swap (int a[], int i, int j) ;
int main () {
  int a[2];
  a[0] = 10；  a[1] = 100;
  printf ("%d, %d\n", a[0], a[1]);
  swap（a, 0, 1）;
  printf ("%d, %d\n", a[0], a[1]);
  return 0;
}
```

```
void swap(int x[], int i, int j) {
  int tmp = x[i];
      x[i] = x[j];
      x[j] = tmp;
}
```

运行结果?

数组名作函数参数

例: 有一维数组score，内放10个学生成绩，求平均成绩。

- 解题思路：
  - 用函数average求平均成绩，用数组名作为函数实参，形参也用数组名
  - 在average函数中引用各数组元素，求平均成绩并返回main函数

```
#include <stdio.h>
int aver;
int main()
{ float average(float array[10]);
   float score[10], aver;  int i;
   printf("input 10 scores:\n");
   for(i=0;i<10;i++)
        scanf("%f",&score[i]);
   printf("\n");
   aver=average(score);
   printf("%5.2f\n",aver);
   return 0;    }
```

```
float average(float array[ ])
{  float aver, sum=array[0];
    for(int i=1;i<10;i++)
       sum=sum+array[i];
    aver=sum/10;
    return(aver);
}
```

```
input 10 scores:
100 56 78 98 67.5 99 54 88.5 76 58

77.50
```

+Chp07_平均成绩.cpp

- 例 有两个班级，分别有35名和30名学生，调用一个average函数，分别求这两个班的学生的平均成绩。
- 解题思路：
  - 需要解决怎样用同一个函数求两个不同长度的数组的平均值的问题
  - 定义average函数时不指定数组的长度，在形参表中增加一个整型变量i
  - 从主函数把数组实际长度从实参传递给形参i
  - 这个i用来在average函数中控制循环的次数
  - 为简化，设两个班的学生数分别为5和10

```
#include <stdio.h>
int main()
{ float average(float array[ ], int n);
   float score1[5]={98,97,91,60,55};
   float score2[10]={67,89,99,69,77,89,76,54,60,99};
   printf("%f\n" , average(score1,5));
   printf("%f\n", average(score2,10));

   return 0;}
```

+Chp07_两班成绩.cpp

调用形式为average(score1,5)时

```
float average(float array[ ], int n)
 { int i;
float aver,sum=array[0];
    for(i=1;i<n; i++)
       sum=sum+array[i];
    aver=sum/n;
    return(aver);
}
```

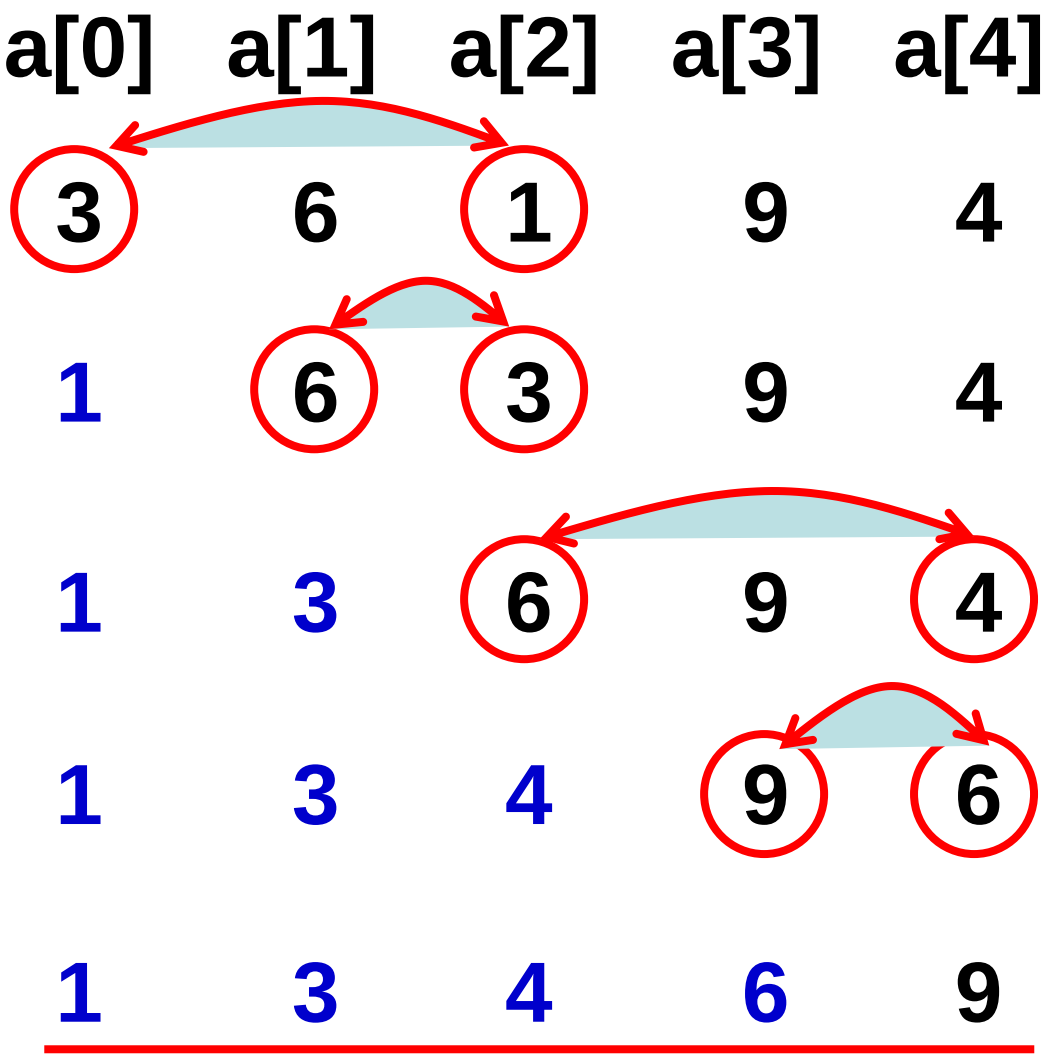
指定多少个数的平均数

调用形式为average(score2,5)时

```
float average(float array[ ], int n)
 { int i;
    float aver,sum=array[0];
    for(i=1; i<n; i++)
       sum=sum + array[i];
    aver=sum/n;
    return(aver);
}
```

指定求10个数的平均数

重温排序



小到大排序

```
#include <stdio.h>
int main()
{ void sort(int array[], int n);
   int a[10],i;
   printf("enter array:\n");
   for(i=0; i<10; i++)  scanf("%d",&a[i]);
   sort(a,10);
   printf("The sorted array:\n");
   for(i=0;i<10;i++)   printf("%d ",a[i]);
return 0;    }
```

+Chp07_数组排序.cpp

```
void sort(int array[], int n)
 { int i, j, k, t;
    for(i=0;i<n-1;i++)
    {  k=i;
       for(j=i+1; j<n; j++)
            if(array[j]<array[k])    k=j;
       t=array[k]; array[k]=array[i]; array[i]=t;
       }
}
```

```
enter array:
45 2 9 0 -3 54 12 5 66 33
The sorted array:
-3 0 2 5 9 12 33 45 54 66
```

快速排序（快排）

#135 身份证排序

按照生日对它们从大到小排序，如果日期相同，则按身份证号码大小排序

```
int main() {
    int n;
    scanf("%d", &n);
    long long id[n];
    for (int i = 0; i < n; i++)
        scanf("%lld", &id[i]);
    qsort(id, n, sizeof(id[0]), compare);
    for (int i = 0; i < n; i++)
        printf("%lld\n", id[i]);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
int compare(const void *a, const void *b) {
    long long int *x = (long long int *)a;
    long long int *y = (long long *)b;
    long long a1 = (*x) % 1000000000000 / 10000;
    long long b1 = (*y) % 1000000000000 / 10000;
    if (b1 > a1)    return 1;
    else if (b1 < a1)    return -1;
        else if (*x > *y)    return -1;
            else if (*x == *y) return 0;
                else    return 1;
}
```

+Chp07_身份证排序(#135).cpp

```
int num[100];
int cmp_int(const void* a , const void* b)
{   int* x = (int*)a;  //强制类型
    int* y = (int*)b;
    return *x - *y; //升序     }
qsort(num, 100, sizeof(num[0]), cmp_int);
```

```
char word[100][10];
int cmp_string(const void* a , const void* b)
{    char* x = (char*)a;
     char* y = (char*)b;
     return strcmp(x, y);     }
qsort(word,100,sizeof(word[0]),cmp_string);
```

```
typedef struct _stu
{   char name[10];
    float score;
} Stu;
```

+Chp07_Qsort结构体.cpp

```
int callBackCompare(const void *a, const void *b)//多排序
{ Stu*x = (Stu*)a; Stu*y = (Stu*)b;
  if(strcmp( (*x).name, (*y).name)>0 ) return 1; //1级升序
  else
       if( (*x).score < (*y).score ) return 1; //2级降序
       else return 0;
}
int main(void)
{   Stu a[] = {  {"aaa",23.5},  {"xxx",45.6}, {"bbb",89},  {"xxx",23.4}  };
    qsort(a, sizeof(a)/sizeof(*a), sizeof(*a), callBackCompare);
   for(int i = 0;i<4;i++)
       printf("%s,%f\n",stu[i].name,stu[i].score);
    return 0;     }
```

中国人民大学
RENMIN UNIVERSITY OF CHINA

3. 全局与局部变量

局部变量

- 在哪里定义变量，有三个地方：
  - 函数的外部定义
  - 函数开头定义
  - 函数内的复合语句内定义
- 在一个函数内部定义的变量只在本函数范围内有效
- 在复合语句内定义的变量只在本复合语句范围内有效
- 在函数内部或复合语句内部定义的变量称为“局部变量”

变量类型

```
int p=1,q=5

float f1(int a)
  { int x, y;   ......  }

char c=1,d=2;

char f2 (int x, int y)
  { int a, b;   ......  }

int main ( )
{  int m, n;
     ......
     return 0;
}
```

全局变量：
p、q、c、d

局部变量：
x, y, a, b, m, n

函数之外定义的变量

```
int main ( )
{   int a, b;
      ......
    { int c;
      c=a+b;
      ......
    }
  ......
}
```

a、b仅在此复合语句内有效

c仅在此复合语句内有效

若外部变量与局部变量同名

```
#include <stdio.h>
int a=3,b=5;
int main()
{ int max(int a,int b);
   int a=8;
   printf("max=%d\n", max(a,b));
   return 0;
}

int max(int a,int b)
{ int c;
   c=a>b?a:b;
   return(c);
}
```

b为全部变量

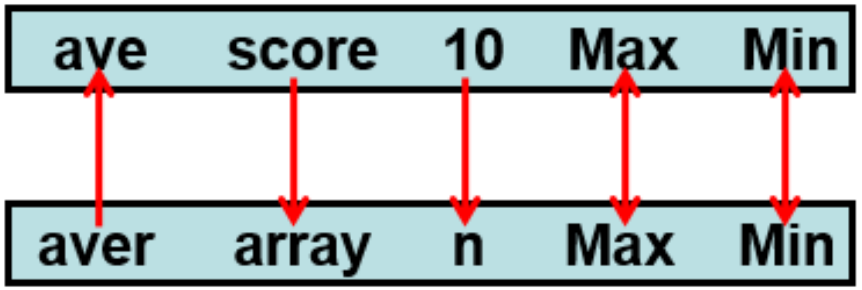
a为局部变量，仅在此函数内有效

+Chp07_全局与局部同名.cpp

例 有一数组，有10个学生成绩，写一个函数，当主函数调用此函数后，能求出平均分、最高分和最低分。

```
#include <stdio.h>
float Max=0,Min=0;
int main()
{ float average(float array[ ],int n);
   float ave, score[10];  int i;
   printf("Please enter 10 scores:\n");
   for(i=0;i<10;i++)
       scanf("%f", &score[i]);
   ave= average(score,10);
   printf("max=%6.2f\nmin=%6.2f\n
         average=%6.2f\n",Max, Min, ave);
   return 0;    }
```

```
float average(float array[ ],int n)
{ int i;  float aver, sum=array[0];
   Max=Min=array[0];
   for(i=1;i<n;i++)
   { if(array[i]>Max) Max=array[i];
      else if(array[i]<Min) Min=array[i];
      sum=sum+array[i];
    }
    aver=sum/n;
    return(aver);
}
```



建议不在必要时不要使用全局变量！！！

```
Please enter 10 scores:
89 95 87.5 100 67.5 97 59 84 73 90
max=100.00
min= 59.00
average= 84.20
```

+Chp07_多项成绩.cpp

#152 确定进制

6*9 = 42 对于十进制来说是错误的，但是对于 13 进制来说是正确的。即, 6(13) * 9(13) = 42(13)， 而 42(13) = 4 * 131 + 2 * 130 = 54(10)。你任务是写一段程序读入三个整数 p、q 和 r，然后确定一个进制 B(2<=B<=16) 使得 p * q = r 成立。如果 B 有很多选择, 输出最小的一个。例如： p = 11, q = 11, r = 121. 则有 **11(3) * 11(3) = 121(3)** 因为 11(3) = 1 * 3^1 + 1 * 3^0 = 4(10) 和 121(3) = 1 * 3 ^ 2 + 2 * 3^1 + 1 * 3^0 = 16(10)。 对于进制 10,有 **11(10) * 11(10) = 121(10)**。这种情况下，应该输出 3。如果没有合适的进制，则输出 0。注意：42(13) = 4 * 13^1 + 2 * 13^0 = 54(10)中 13^1 表示 13 的一次方，13^0 表示 13 的 0 次方，依此类推。

输入格式： 输入有 T(T<=100)组测试样例。 T 在第一行给出。每一组测试样例占一行，包含三个整数 p、q、r。 p、q、r 的所有位都是数字，并且 1 <= p、q、r <= 10,000,000

输出格式： 对于每个测试样例输出一行。该行包含一个整数：即使得 p * q = r 成立的最小的 B。如果没有合适的 B，则输出 0。

输入样例

```
3
6 9 42
11 11 121
2 2 2
```

输出样例

```
13
3
0
```

6(13) * 9(13) = 42(13)

```
int Transf(int a[100][3], int i, int j, int t) {//a 数组第 i 行第 j 个数 t 进制转换 10 进制
    int x, d, sum, zs;
    for(sum=0, x=a[i][j], zs=1; x>0; x/=10, zs*=t) {
        d=x%10;
        if (d>=t) {sum=-1; break;} }
        sum+=d*zs;
    return sum;
}
int GetMinBase(int a[100][3], int i) {
    int p, q, r, t, ok=0;
    for (t=2; t<=16; t++) {
        p=Transf(a, i, 0, t); q=Transf(a, i, 1, t); r=Transf(a, i, 2, t);
        if (p==-1 || q==-1 || r==-1) continue;
        if (p*q==r) {ok=1; break;}    }
    return ok==1? t : 0;
}
int main() {
    int n, a[100][3];
    scanf("%d", &n);
    for (int i = 0; i<=n-1; i++) {
        scanf("%d%d%d", &a[i][0], &a[i][1], &a[i][2]);
        printf("%d\n", GetMinBase(a, i));    }
    return 0;
}
```

数组a中第i组数的第j个数看作是t进制转换成十进制

数组a中第i组数的最小进制

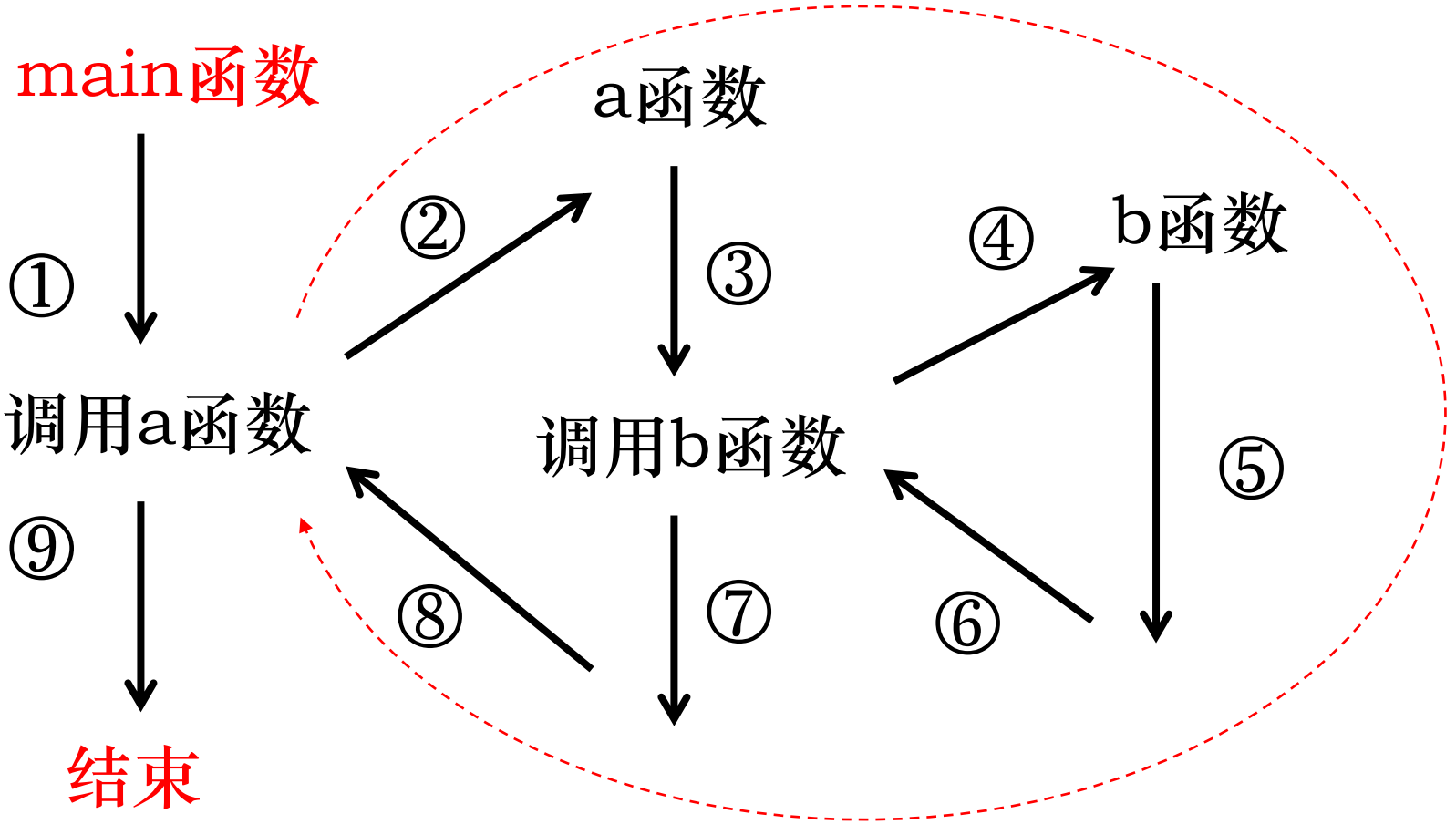
+Chp07_确定进制(#152).cpp

数组a[100][3]设置为全局变量

中国人民大学
RENMIN UNIVERSITY OF CHINA

4. 嵌套调用

函数的嵌套调用



函数的嵌套调用

主函数

输入4个整数，找出其中最大的数

```
#include <stdio.h>
int main()
{ int max4(int a,int b,int c,int d);
   int a,b,c,d,max;
   printf("4 interger numbers:");
   scanf("%d%d%d%d",&a,&b,&c,&d);
   max= max4(a,b,c,d);
   printf("max=%d \n",max);
   return 0;   }
```

+Chp07_Max嵌套调用.cpp

max4函数

```
int max4(int a, int b, int c, int d)
{  int max2(int a, int b);
    int m;
    m=max2(a, b);
    m=max2(m, c);
    m=max2(m, d);
    return(m);
}
```

ruturn max2(max2(max2(a,b),c),d);

```
int max2(int a, int b) {
return(a>b?a:b);   }
```

```
#include <stdio.h>
int main()
{  ……
    max=max4(a,b,c,d);
   ……
}
```

```
int max4(int a,int b,int c,int d)
{  int max2(int a,int b);
    ruturn max2(max2(max2(a,b),c),d);
}
```

```
int max2(int a,int b) {
return(a>b?a:b);   }
```

```
4 interger numbers:12 45 -6 89
max=89
```

5. 模块化编程举例

#304 整数计数

编写一个程序计算整数区间[a, b]内，其个位数是 n，且能被 k 整除的 m 位正整数共有多少个。

【输入格式】

输入只有一行，输入 5 个整数 a、b、n、k、m，空格分隔，其中：1≤a≤b≤1,000,000，且 0≤n, k, m≤9。

【输出格式】

输出一行，为符合要求的整数个数。

【样例输入 1】

1019 1 2 2

【样例输出 1】

0

【样例输入 2】

1050 4 2 2

【样例输出 2】

4

```
int main()
{
    int a,b,c,d,e,s=0,i,f;
    scanf("%d%d%d%d%d",&a,&b,&c,&d,&e);
    f=(pow(10,e-1));
    for(i=a;i<=b;i++)
        if(i%10==c)
            if(i%d==0)
                if(i/(10*f)<1&&i/f>=1)  s++;
    printf("%d",s);
    return 0;
}
```

+Chp07_整数计数(#304).cpp

#同构数

```
int main() {
    int from, to, n, m, weishu, sum=0, i;
    scanf("%d%d", &from, &to);
    for (n = from; n <= to; ++n) {
        i = n;
        weishu = 0;
        do {i = i / 10;
            weishu++;
        } while (i != 0);
        m = n * n;
        for (i = 0; i < weishu; i++)
            m = m / 10;
        for (i = 0; i < weishu; i++)
            m = m * 10;
        if (n == n * n - m)  sum += n;
    }
    printf("%d", sum);
    return 0;
}
```

+Chp07_同构数(#155).cpp

```
int ismp(int x){
    int sq = x*x;  int t= x;  int base= 1;
    while(t!=0){
        t /= 10;  base *= 10; }
    if(sq%base == x) return 1;
    else return 0;
}

int main(){
    int a,b;
    scanf("%d%d",&a,&b);
    int sum=0;
    for(int i=a; i<=b; i++)
        if(ismp(i)) sum += i;
    printf("%d",sum);
    return 0;
}
```

#291 大整数加减法

```
void parseStrToIntArr(char str[],int a[],int size)
{  for(int i=0;i<size;i++)
      a[i]=str[size-1-i]-'0';
}
int bigIntMinus(int m[],int a[],int b[],int len)
{for(int i=0;i<len;i++)
    {  m[i]=a[i]-b[i];
       if(m[i]<0)  {a[i+1]--;  m[i]+=10;}
    }
    while(len>1&&m[len-1]==0)len--;
    return len;
}
int add(int s[],int a[],int b[],int len)
{  for(int i=0;i<len;i++)
    {s[i]+=a[i]+b[i];
       if(s[i]>9){s[i+1]+=s[i]/10; s[i]%=10;}
    }
    if(s[len])len++;
    return len;
}
```

```
int main()
{
    char ac[2000],bc[2000];
    int an[2000]={0},bn[2000]={0};
    char op, s1=1,s2=1;
    cin>>op;
    cin>>ac;
    cin>>bc;
    if(ac[0]=='-')
    {   s1=-1;
        strcpy(ac,&ac[1]);//将 ac 去除第一位,
    }
    if(bc[0]=='-')
    {   s2=-1;
        strcpy(bc,&bc[1]);
    }
```

#291 大整数加减法

```
int lena=strlen(ac);
int lenb=strlen(bc);

parseStrToIntArr(ac,an,lena);
parseStrToIntArr(bc,bn,lenb);

int len=(lena>lenb)?lena:lenb;

if(op=='-')s2*=-1;
int sum[2018]={0};
```

```
if(s1==s2)
{   len=add(sum,bn,an,len);
    if(s1==-1)cout<<"-";
}
else
    if((lena>lenb)||(lena==lenb&&strcmp(ac,bc)>0))
        {len=bigIntMinus(sum,an,bn,len);
        if(s1==-1)cout<<"-";}
    else
        {len=bigIntMinus(sum,bn,an,len);
        if(s2==-1)cout<<"-";}
for(int i=len-1;i>=0;i--)
    cout<<sum[i];
```

+Chp07_大整数加减法(#291).cpp

#301 成绩统计

期中考试后，老师需要对班里学生的成绩进行统计分析，了解各个分数段的人数。给定所有学生的成绩和成绩分段方式，请你编程帮忙**以下任务：**

1）统计各个分数段的人数，并按照分数段人数从大到小输出；

2）将按分数段统计的结果用模拟的直方图显示出来。

【输入格式】

第 1 行包含 2 个整数；n、m、g，分别表示学生人数、分数段个数、需解决的任务编号。

m 表示将成绩区间[0,100]平均分成 m 个分数段。例如 m = 5 表示分为 5 个分数段，分别是[0,20)、[20,40)、[40,60)、[60,80)、[80,100]。

g 一共有 3 种取值，分别是 0、1、2，

- 0 表示任务 1、2 均需完成，且先完成任务 1，再完成任务 2；
- 1 表示只完成任务 1；
- 2 表示只完成任务 2。

第 2 行包含 n 个[0,100]之间的整数，为 n 个学生的期中考试成绩。

+Chp07_成绩统计(#301).cpp

【输出格式】

任务 1 格式：

若干行，每行一个分数段以及该分数段的人数，分数段表示为 **[L,R)**或者[L,100]的形式，其中 L 占 2 个字符的宽度，R 需要占用 3 个字符的宽度，分数段后面跟一个字符’：’，之后输出一个空格，再输出该分数段的人数。

注意：如果该分数段的人数为 0，则不输出该分数段；按照分数段的人数从大到小的顺序输出，如果 2 个分数段人数相同，先输出分数段比较低的那一个。

任务 2 格式：

分数段的输出格式与任务 1 相同，’：’之后输出若干个字符’*’，具体个数为该分数段的实际人数，或者该分数段人数归一化后的数量。人数最多的分数段能在显示在一行以内，**归一化的方式**为人数最多那个分数段最多输出 50 个字符’*’，即如果人数多于 50，则输出 50 个字符’*’，其他分数段按比例减少字符’*’的个数（下取整）；否则直接输出实际数量的字符’*’。

注意：任务 1 和任务 2 之间请输出一个空行。

【输入样例 1】

20 10 0

88 90 75 68 77 85 86 99 10095 60 55 32 93 63 60 78 96 70 80

【输出样例 1】

[90,100): 6

[60, 70): 4

[70, 80): 4

[80, 90): 4

[30, 40): 1

[50, 60): 1

[0, 10):

[10, 20):

[20, 30):

[30, 40):*

[40, 50):

[50, 60):*

[60, 70):****

[70, 80):****

[80, 90):****

[90,100):******

```
#include <stdio.h>
#include <math.h>
int main() {
    int n, m, g, i, j, sc[5001], a, b, c, d, t = 0, k[11] = {0}, ch[11][3];
    scanf("%d %d %d", &n, &m, &g);
    for (i = 0; i < n; i++)
        scanf("%d", &sc[i]);
    a = 100 / m;
    for (i = 0; i < n; i++) {//每个分数
        for (j = 0; j < m; j++)//每个分数段
            if (sc[i] >= j * a && sc[i] < (j + 1)*a)
                k[j] += 1;
        if (sc[i] == 100)  k[m - 1] += 1;
    }
    for (i = 0; i < m; i++) {
        ch[i][0] = i;
        ch[i][1] = k[i];
    }
    for (i = 0; i < m; i++)//统计结果排序
        for (j = 0; j < m; j++)
            if ((ch[i][1]>ch[j][1])||(ch[i][1]==ch[j][1] && ch[i][0]<ch[j][0])) {
                c = ch[i][1];  ch[i][1] = ch[j][1];  ch[j][1] = c;
                d = ch[i][0];  ch[i][0] = ch[j][0];  ch[j][0] = d;
            }
```

```
if (g == 1)
    for (i = 0; i < m; i++)
        if (ch[i][1] != 0) {
            if (ch[i][0] == 0)
                printf("[0,%d)", (ch[i][0] + 1)*a);
            else
                if ((ch[i][0] + 1)*a != 100)
                    printf("[%d,%d)", ch[i][0]*a, (ch[i][0] + 1)*a);
                else
                    printf("[%d,100]", ch[i][0]*a);
            printf(":%d", ch[i][1]);
            printf("\n");
        }
```

```
if (g == 0) {
    for (i = 0; i < m; i++)
        if (ch[i][1] != 0) {
            if (ch[i][0] == 0)
                printf("[0,%d)", (ch[i][0] + 1)*a);
            else
                if ((ch[i][0] + 1)*a != 100)
                    printf("[%d,%d)", ch[i][0]*a, (ch[i][0] + 1)*a);
                else
                    printf("[%d,100]", ch[i][0]*a);
            printf(":%d", ch[i][1]);
            printf("\n");
        }

    printf("\n");
    if (ch[0][1] > 50)
        for (j = 0; j < m; j++)
            k[j] = k[j] * 50 / ch[0][1];

    for (i = 0; i < m; i++) {
        if (i == 0)  printf("[0,%d)", a);
        else
            if (i != m - 1)  printf("[%d,%d)", i * a, (i + 1)*a);
            else  printf("[%d,100]", i * a);
        printf(":");
        for (j = 0; j < k[i]; j++)  printf("*");
        printf("\n");
    }
}
```

输出结果

画图

```
if (g == 2) {
    if (ch[0][1] > 50)
        for (j = 0; j < m; j++)
            k[j] = k[j] * 50 / ch[0][1];

    for (i = 0; i < m; i++) {
        if (i == 0)  printf("[0,%d)", a);
        else
            if (i != m - 1)  printf("[%d,%d)", i * a, (i + 1)*a);
            else  printf("[%d,100]", i * a);
        printf(":");
        for (j = 0; j < k[i]; j++)  printf("*");
        printf("\n");
    }
}
```

```
int n, m, g, i, j, sc[5001], a, b, c, d, t = 0, k[11] = {0}, ch[11][3];
int main() {
    void print_result();
    void print_figure();
    scanf("%d %d %d", &n, &m, &g);
    for (i = 0; i < n; i++)
        scanf("%d", &sc[i]);
    a = 100 / m;
    for (i = 0; i < n; i++) {//每个分数
        for (j = 0; j < m; j++) //每个分数段
            if (sc[i] >= j * a && sc[i] < (j + 1)*a)  k[j] += 1;
        if (sc[i] == 100)  k[m - 1] += 1;
    }
    for (i = 0; i < m; i++) {
        ch[i][0] = i;     ch[i][1] = k[i]; }
    for (i = 0; i < m; i++) //统计结果排序
    for (j = 0; j < m; j++)
    if ((ch[i][1]>ch[j][1])||(ch[i][1]==ch[j][1]&&ch[i][0]<ch[j][0]))
        {c = ch[i][1];  ch[i][1] = ch[j][1];  ch[j][1] = c;
        d = ch[i][0];  ch[i][0] = ch[j][0];  ch[j][0] = d;}

    if (g == 1) print_result();
    if (g == 2) print_figure();
    if (g == 0) {print_result(); print_figure();}
    return 0;
```

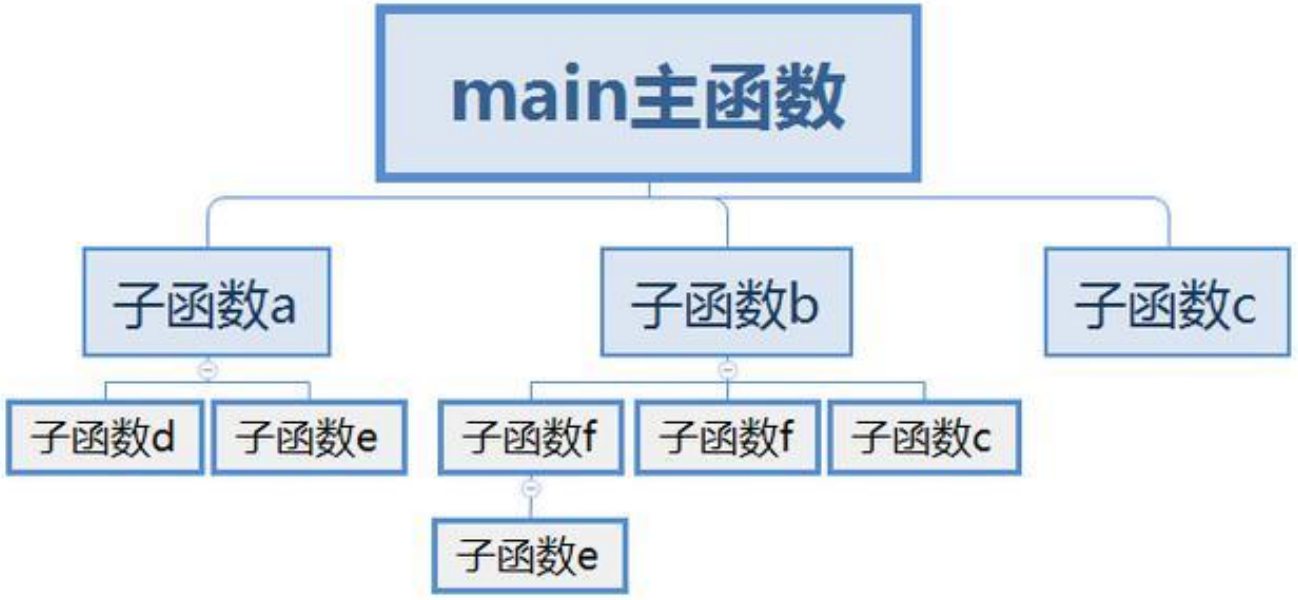
```
void print_result()
  {for (i = 0; i < m; i++)
   if (ch[i][1] != 0) {
       if (ch[i][0] == 0)
          printf("[0, %d)", (ch[i][0] + 1)*a);
       else
          if ((ch[i][0] + 1)*a != 100)
             printf("[%d, %d)", ch[i][0]*a, (ch[i][0]+1)*a);
          else
             printf("[%d,100]", ch[i][0]*a);
    printf(":%d", ch[i][1]);
    printf("\n");      }
  }
void print_figure()
  {if (ch[0][1] > 50)
     for (j = 0; j < m; j++)
        k[j] = k[j] * 50 / ch[0][1];

  for (i = 0; i < m; i++) {
   if (i == 0)  printf("[0, %d)", a);
   else
     if (i!=m-1) printf("[%d, %d)",i*a, (i+1)*a);
     else     printf("[%d,100]", i * a);
    printf(":");
    for (j = 0; j < k[i]; j++)    printf("*");
        printf("\n");   }
}
```

总结

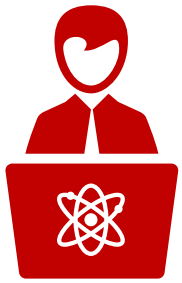
- 函数基本概念
- 数组作为函数参数*
- 外部变量与内部变量*
- 函数嵌套调用*
- 应用实例

面向过程的编程语言：
函数是最高程序组织单元

面向对象的编程语言：
对象是最高程序组织单元



中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！