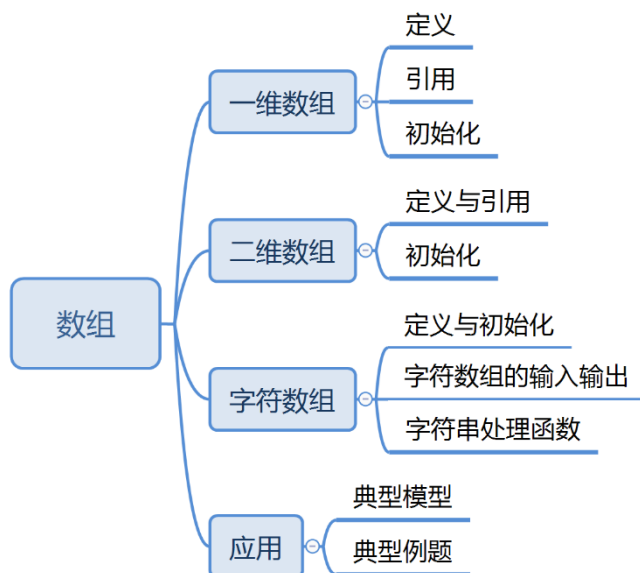




专题五 数组

【内容预览】



5.1、数组

5.1.1、数组的定义

数组是一个在内存中顺序排列的、由若干**相同数据类型**的元素组成的数据集合，其所有元素共用一个名字，每个元素用带方括号的序号（称为下标）来区分，下标实际上就是元素在数组中的位置值。

5.2、一维数组

【知识清单】

5.2.1、一维数组定义

定义格式：

类型说明符 数组名[整型常量表达式]

如： `int a[10];`

1. 数组名代表该数组在内存中的首地址，是一个地址**常量**（因为数组名是常量，这也**照应了后面的“如果数组已经定义，则不能直接使用赋值语句直接将一个字符串常量或字符数组直接赋给另一个字符数组”**）

如： `char a[10];`

`a[10] = "abcde";` //这个就是错误的

2. []中**不允许有变量**（旧版 C99 标准不允许，新版 C11 标准允许，国内大多数教材按照 C99 编写，同学

们请按照老师的要求来答题), 即使该变量已经定义且有初值也不允许, 不过为了适应数组的变化, 常用的方法是用符号常量来指定元素的个数 (1) `int n;` (2) `int n=20;`

```
scanf("%d",&n);    int a[n];
int a[n];           //这两种方法都是错误的
```

但是[]中可以有预处理中的宏定义, 例如:

```
#define N 5    //N 就代表了 5
.....
int a[N];
```

5.2.2、数组元素的引用

引用格式: 数组名[下标] 如: `a[10]`

1. 下标值从 0 开始, 对于一个具有 `n` 个元素的一维数组来说, 它的下标值是 0 到 `n-1`。
2. 使用数组元素的过程中, 下标可以是常量, 常量表达式, 变量, 变量表达式, 甚至是另一个数组元素。
如: `a[1+2], score[i], score[i+3], a[a[3]]`

5.2.3、数组的初始化

数组的初始化是指在定义数组时对其中的全部或部分元素指定初始值。

1. 全部赋初值
如: `int a[]={0,2,4,6,8,10};` 全部赋初值可以省略下标
2. 部分赋初值
如: `int b[15]={1,3,5,7,9};`
系统会为前几个元素赋初值, 而剩下的数组元素赋初值 0; 且部分赋初值的情况, 下标不能省略。

5.2.4、数组的常用算法应用

1. 查找: 指从若干相关数据中找到指定数据。
2. 排序: 冒泡排序法 (重点)
3. 插入: 方法一: 先查找位置, 再放置数据
将要插入的数与每个元素比较, 找插入的位置, 依次将插入点后的元素向后移动
方法二: 边查找边移动
4. 删除: 将后面的元素前移覆盖该元素
5. 统计: 统计一组数据中特定数据出现的次数。

5.2.5、典型模型分析

1. 用选择排序法将数组从小到大排序
思路: 先进行一次排序, 找到一个基准数, 认为是最小的数, 然后依次比较, 找到并交换比它小的数。然后再进行第二遍排序, 找到除第一个数外最小的数。依此类推, 直到将 `N` 个元素排序。

```
#include<stdio.h>
#define N 5
void main()
{
    int i,j,t;
    int a[N];
    for(i=0;i<N;i++)    //输入数组中的元素
        scanf("%d",&a[i]);
    for(i=0;i<N-1;i++)    //进行 N-1 遍排序
        for(j=i+1;j<N;j++)    //对剩下的进行搜索
            if(a[i]>a[j])    //剩下的元素和第一个元素进行比较, 如果比第一个元素小,
```

将最小的元素和第一个交换

```

    {
        t=a[i];
        a[i]=a[j];
        a[j]=t;
    }
    for(i=0;i<N;i++)        //输出排序后的数组
        printf("%d\t",a[i]);
}

```

2. 用冒泡排序法将数组从小到大排序

思路：从第一个元素开始，将相邻的两个数进行比较，将小的数字放在前面，大的放在后面，依次往后比较，比较到最后一组后，产生一个最大值。第 n 遍排序是去除产生的最大值，对剩下的序列进行类似第一遍冒泡排序。

```

#include<stdio.h>
#define N 5
void main()
{
    int i,j,t;        //定义变量和数组
    int a[N];
    for(i=0;i<N;i++) //依次输入 5 个数
        scanf("%d",&a[i]);
    for(i=0;i<N-1;i++) //进行 N-1 遍排序
        for(j=0;j<N-1-i;j++) //对剩下的数据进行冒泡排序，
                                //将该次最大值放到上一次排序最大值之前
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
    for(i=0;i<N;i++) //输出排序后的数组
        printf("%d\t",a[i]);
}

```

【解题技巧】

例 5.2.1 以下能对一维数组 a 进行正确初始化的语句是 ()

A) `int a[5]={0, 0, 0, 0, 0};`

B) `int a[]={};`

C) `int a[]={0};`

D) `int a[5]={1,1,1,1,1};`

正解：A、C

分析：B 没有给出数组的大小，D 中数组只能容纳 5 个元素，却赋初值有 6 个元素。

5.3、二维数组

【知识清单】

5.3.1、定义与引用

引用方法：数组名[行下标][列下标]

5.3.2、初始化

方式一：按行初始化 指在初始化表中每一对花括号对应一行的元素

如：int a[2][3]={ {1,2,3}, {4,5,6} };

方式二：按存放顺序初始化 指按内存中的存放顺序将初始值分别赋值给对应元素

如：int a[2][3]={1,2,3,4,5,6};

在初始化二维数组时可省略第一维大小，但第二维的大小不能省略。

如：int a[2][]={1,2,3,4,5,6}; 就是错误的。

5.3.3、典型例题

1. 有一个 3×3 矩阵，要求编程求出最大的那个元素的值，以及其所在的行和列

```
#include<stdio.h>
void main()
{
    int i,j,max;
    int row=0,column=0;
    int a[3][3]={ {1,2,3}, {2,-3,4}, {9,4,7} };
    max=a[0][0];
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            if(a[i][j]>=max)
            {
                max=a[i][j];
                row=i;
                column=j;
            }
    printf("max=%d,row=%d,column=%d\n",max,row,column);
}
```

2. 输入一个 4×4 整数矩阵，然后将之转置并显示这个转置后的矩阵

```
#include<stdio.h>
#define SIZE 4
void main()
{
    int data[SIZE][SIZE],i,j,d;
    for(i=0;i<SIZE;i++)
        for(j=0;j<SIZE;j++)
```

```

        scanf("%d",&data[i][j]);    //输入矩阵的值
for(i=1;i<SIZE;i++)                //矩阵转置
    for(j=1;j<=i;j++)
    {
        d=data[i][i-j];
        data[i][i-j]=data[i-j][i];
        data[i-j][i]=d;            //交换所有对称点元素
    }
for(i=0;i<SIZE;i++)
{
    printf("\n");
    for(j=0;j<SIZE;j++)
    {
        printf("%4d",data[i][j]);
    }
}
}

```

3. 计算两个矩阵的乘积

```

#include <stdio.h>
void main()
{
    int m,n,q,i,j,k,s=0,t=0,a[9][9],b[9][9],c[9][9];
    printf("请输入 a 矩阵的行数 m 和列数 n 以及 b 矩阵的行数 n 和列数 q:\n");
    scanf("%d%d%d",&m,&n,&q);
    printf("请输入 a 矩阵的 m*n 个数: \n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);    //输入 a 矩阵的元素
    }
    printf("请输入 b 矩阵的 n*q 个数:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<q;j++)
            scanf("%d",&b[i][j]);    //输入 b 矩阵的元素
    }
    for(i=0;i<m;i++)
        for(j=0;j<q;j++)
        {
            for(k=0;k<n;k++)
            {
                t=a[i][k]*b[k][j];
                s=s+t;
            }
            c[i][j]=s;                //利用两个矩阵乘积的公式一步步将新的矩阵赋给 c 矩阵
            s=0;
        }
    }
}

```

```

    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<q;j++)
            printf("%6d",c[i][j]);    //输出结果矩阵
        printf("\n");
    }
}

```

4. 寻找矩阵的鞍点（该位置上的元素是该行上的最大值，同时也是该列上的最小值）

```

#include<stdio.h>
#define N 3
#define M 4
int main()
{
    int a[N][M],    max,min,i,j,k;    //定义一个 3×4 的矩阵

    printf("请输入数组: \n");
    for(i=0;i<N;i++)                //输入矩阵中各个元素的值
        for(j=0;j<M;j++)
            scanf("%d",&a[i][j]);
    for(i=0;i<N;i++)
    { for(j=0;j<M;j++)
        printf("%5d",a[i][j]);
        printf("\n");
    }
    for(i=0;i<N;i++)
    {
        max=a[i][0];                //先以每行第一个为基准
        for(j=1;j<M;j++)
            {if(max<a[i][j])
                {max=a[i][j];
                    k=j;            //找到行中最大的元素
                }
            }
        min=a[0][k];
        for(j=1;j<N;j++)
            { if(min>a[j][k])
                min=a[j][k];        //找到列中最小的元素
            }
        if(min==max)                //看每行最大和每列最小是否是同一个数
            printf("鞍点为: %d\n",max);
    }
    printf("搜索结束");
    return 0;
}

```

【解题技巧】

例 5.3.1 下列数组的定义正确的是 ()

- A)int a[5];
- B)int a[2][]={1,2,3,4,5};
- C)int n=20;
int a[n];
- D)int n;
scanf("%d",&n);
int a[n];

正解: A
分析: 二维数组初始化时,可省略第一维大小,但第二维大小不能省略;
定义数组是[]中不允许有变量,即使该变量已经定义且有初值也不允许;

例 5.3.2 下列二维数组定义正确的是 ()

- A)int a[2][3]={1,2},{2,3},{3,4}};
- B)int a[][3]={1,2,3,4,5,6};
- C)int a[2][]={1,2,3,4,5,6};
- D)int a[2][]={{1,2},{3,4}};

正解: B
分析: 定义二维数组,第二维的大小必须定义,因此排除 C、D。A 中定义的数组行数只有 2,但是赋的初值却有 3 行。

5.4、字符数组

【知识清单】

5.4.1、定义及初始化

初始化 方法一: 逐个字符进行初始化。
方法二: 用字符串进行初始化。
C 语言中,字符串在内存中连续存储,每个字符占用一个字节,在串的尾部有一个字符串结束字符'\0',那么,一个字符串在内存中存储所占的字节数要比字符串中的字符个数多一个。因此,在定义存放字符串的字符数组时,数组的大小必须大于字符串中的字符个数。
如: char str[5]= "hello";就是错误的。

字符数组不同初始化方式的比较

字符数组的初始化	说明
char str[6]="hello"; char str[6]={ "hello" }; char str[]="hello";	<div>■ 三种方式等价</div> <div>■ 数组 str 中只能存放最多 5 个字符的字符串</div> <div>■ 字符串两边的大括号可以省略</div>
char str[5]={'H','e','l','l','o'}; char str[]={ 'H','e','l','l','o' };	<div>■ 两种方式等价</div> <div>■ 初始化后,字符数组中没有字符串结束符'\0'</div>

```
char str[10]="hello";
char str[10]={'H','e','l','l','o'};
```

- 两种方式等价
- 多余的元素被自动赋值为'\0'(即 0)

注意 `char str[5]={ "hello"}` 是错误定义，因为字符串 `"hello"` 占用 6 个字节内存空间，而字符数组只定义了 5 个字节的内存空间。

数组的初始化是指在定义数组的同时给数组元素赋初值，如果数组已经定义，则不能使用赋值语句直接将一个字符串常量或字符数组直接赋给另一个字符数组。

如：`char a[10],b="hello";`

```
a[]="abcde";    //错误
a="abcde";      //错误
a=b;            //错误
```

5.4.2、字符数组的输入与输出

1. 逐个输入/输出字符数组中的字符

(1) 在 `scanf()/printf()` 函数中使用 `"%c"` 格式 `scanf("%c",&name[i]);`
`printf("%c",name[i]);`

(2) 用 `getchar()/putchar()` 函数 `name[i]=getchar();putchar(name[i]);`

2. 输入/输出整个字符串

(1) 在 `scanf()/printf()` 函数中使用 `"%s"` 格式符 `scanf("%s",name);`
`printf("%s",name);`

a) 字符数组名是地址常量，因此 `scanf` 函数输入项直接写数组名，不用加地址符 `&`。

b) `scanf()` 函数是以空格、制表符、换行符和文件结束符 `EOF` 作为输入的结束标志，因此，用 `"%s"` 格式符输入时不能包含这些字符。

如：对数组 `name` 执行时输入 `John Smith`，由于在中间有空格，系统自认为输入结束，仅将 `John\0` 放到 `name` 中。

(2) 使用 `gets()/puts()` 函数

格式：`gets(字符数组名)` `char name[15]; gets(name); puts(name);`

功能：将键盘上接收到的一个字符串存放到所指定的字符数组中

`gets` 和 `puts` 函数以换行符或文件结束符 `EOF` 作为输入的结束标志，因此这种输入方式能输入含空格符的字符串。

5.4.3、常用的字符串处理函数

在使用字符串输入输出函数 `gets()` 和 `puts()` 时，应包含在头文件 `"stdio.h"` 里，而使用其它字符串处理函数，则应包含头文件 `"string.h"`。

1. 测试字符串长度函数 `strlen`

格式：`strlen(str)`

功能：返回字符串 `str` 的实际长度，不包含 `'\0'` 在内的实际字符的个数

2. 字符串连接函数 `strcat`

格式：`strcat(str1,str2)`

功能：把字符串 `str2` 中的字符串连接到字符串 `str1` 的末尾，将连接后的结果放在字符串 `str1` 中，字符串 `str2` 不变，函数值返回字符串 `str1` 的首地址。

函数有两个参数，`str1` 只能是字符串组名，`str2` 可以是字符串组名，也可以是字符串常量。

3. 字符串复制函数 `strcpy`

格式：`strcpy(str1,str2)`

功能：类比 `strcat` 函数

(1) 类比 `strcat` 函数

(2) 复制时连同 `str2` 中的 `'\0'` 一起被复制到 `str1` 中

(3) 字符数组之间的赋值只能用 `strcpy(str1,str2)`, 不能用 `str1=str2`

4. 字符串比较函数 `strcmp`

格式: `strcmp(str1,str2)`

功能: 比较字符数组 `str1` 和 `str2` 的大小, 函数返回值分三种情况:

$$\begin{cases} \text{str1 等于 str2 则函数返回值等于 0;} \\ \text{str1 大于 str2 则函数返回值大于 0;} \\ \text{str1 小于 str2 则函数返回值小于 0;} \end{cases}$$

(1) 比较两个字符串的大小, 不能直接使用关系运算符, 必须使用函数 `strcmp()`

(2) 两个参数既可以是字符数组, 也可以是字符串。

(3) 比较方法是对两个字符串从左至右按字符的 ASCII 码值大小逐个比较, 直到出现不同的字符或遇到 '\0' 为止。

5. 字母大小写转换函数 `strlwr` 和函数 `strupr`

(1) `strlwr` 函数

格式: `strlwr(str)`

功能: 将字符串中的所有大写英文字母转换成小写, 所有的非大写英文字母不变。

(2) `strupr` 函数

类比上面的函数表述。

5.4.4、典型例题

有 3 个字符串, 要求排序输出

```
#include<stdio.h>
#include<string.h>
void main()
{
    char a[5][10],temp[10];    //定义一个字符串数组和一个用来交换的字符数组
    int i,j;
    for(i=0;i<3;i++)
        gets(a[i]);           //输入 3 个字符串
    for(i=0;i<3;i++)
        for(j=i+1;j<3;j++)
            if(strcmp(a[i],a[j])>0) //3 个字符串依次比较, 将小的字符串放在前面。
                { strcpy(temp,a[i]);
                  strcpy(a[i],a[j]);
                  strcpy(a[j],temp);}
    for(i=0;i<3;i++)           //依次输出 3 个字符串
        puts(a[i]);
}
```

【解题技巧】

例 5.4.1 设有如下定义,

```
char a[]={"Thankyou "};  
char b[]={'T','h','a','n','k','y','o','u'};  
用 strlen 函数求得字符串的长度;
```

则正确的叙述为 ()

- A) 数组 a 和数组 b 等价
- B) 数组 a 和数组 b 的长度相等
- C) 数组 a 的长度大于数组 b 的长度
- D) 数组 a 的长度小于数组 b 的长度

正解: B

分析: strlen 函数返回字符串的实际长度, 不包含 '\0' 在内的实际字符的长度。

例 5.4.2 下面程序段执行后的输出结果是()

```
char *sp="AB\t12\nxy\\0is15\n";
```

```
printf("%d ",strlen(sp));
```

- A) 9 B) 12 C) 14 D) 15

正解: A

分析: 在字符串初始化的时候, 只能赋给 *sp="AB\t12\nxy\\", '\0' 是结束字符, 所以字符串的长度为 9;

例 5.4.3 为了判断两个字符串是否相等, 应当使用()

- A) if(s1==s2)
- B) if(s1=s2)
- C) if(strcmp(s1,s2))
- D) if(strcmp(s1,s2)==0)

正解: D

分析: 字符串不能直接进行 “=” 赋值, 也不能直接进行 “==” 或 “!=” 的比较, 如果需要赋值, 应该借助 strcpy 函数。如果需要比较, 应该借助 strcmp 函数。

【精选习题】

答案 P115

基础篇

1. 有以下语句, 下面 () 是正确的描述 (长度指通过 strlen 函数求得)

```
char x[]="12345";char y[]={ '1','2','3','4','5'};
```

- A) x 数组和 y 数组的长度相同
- B) x 数组长度大于 y 数组长度
- C) x 数组长度小于 y 数组长度
- D) x 数组等价于 y 数组

2. 有两个字符数组 a、b,则以下正确的输入语句是 ()

A) gets(a,b);
B) scanf("%s%s",a,b);
C) scanf("%s%s",&a,&b);
D) gets("a"),gets("b");

3. 下面程序输出的结果是 ()

```
#include<stdio.h>
void main()
{
    int n[3][3],i,j;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            n[i][j]=i+j;
    for(i=0;i<2;i++)
        for(j=0;j<2;j++)
            n[i+1][j+i]+=n[i][j];
    printf("%d\n",n[i][j]);
}
```

A) 14 B) 7 C) 6 D) 不确定

4. 下列数组定义, 不正确的是 ()

A) int a[1][3]; B) int x[2][2]={1,2,3,4};
C) int m[][3]={1,2,3,4,5,6}; D) int x[2][]={1,2,4,6};

5. 下面对数组的定义合法吗? _____

```
int a[255];
a['A']=0;
```

6. 执行以下程序后, 输出的'#'号的个数是_____

```
#include<stdio.h>
void main()
{
    int i,j;
    for(i=1;i<5;i++)
        for(j=i;j>0;j--)
            putchar('#');
}
```

7. 假定一个 int 型变量占用四个字节, 若有定义: int x[10]={0,2,4}; 则数组 x 在内存中所占字节数是 ()

A) 3 B) 6 C) 10 D) 40

8. 若有说明: int a[][4]={1,2,3,4,5,6,7,8,9,10,11,12};, 则数组第一维的大小为: ()

A) 2 B) 3 C) 4 D) 不能确定的值

9. 若有以下说明语句, 则输出结果是()

```
char sp[]="\t\b\0123\n";

printf("%d",strlen(sp));
```

A) 11 B) 7 C) 6 D) 5

10. 下列程序运行的结果为 ()

```
#include<stdio.h>
#include<string.h>
```

```
main()
{
int i;
char a[]="How are you!";
for(i=0;a[i];i++)
if(a[i]==' ')
strcpy(a,&a[i+1]);
printf("%s\n",a);
}
```

A)are you!

B)Howareyou!

C)areyou!

D)you!

提高篇

1. () 读程序, 写出结果

```
#include<stdio.h>
main()
{ char s[]="011230 ";
  int k;
  char c;
  for(k=1;(c=s[k])!= '\0';k++){
    switch(c){
      case '0':putchar('%');continue;
      case '1':putchar('*');
      case '2':putchar('&');continue;
    }
    putchar('#');
  }
```

结果为: _____

2. 有以下程序运行后的输出结果是 ()

```
int main()
{ char a[10]={ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', '\0', 'i'};
  int j;
  j=strlen(a);
  printf("%d",j);
  return 0;
}
```

A)7

B)8

C)9

D)10

3. 若运行时输入: 求程序的输出结果

```
#include<stdio.h>
#include<string.h>
void main()
{ char st[20]= "hello\0\t\\";
  printf("%d %d\n",strlen(st),sizeof(st));
}
```

结果为: _____

4. 输入一串字符，直到输入一个星号（*）为止，统计（输出）其中字母个数和数字字符个数。
5. 求一个 5×5 的矩阵两条对角线的元素和。
6. 计算一个 3×2 的矩阵和一个 2×3 矩阵的乘积。
7. 编程将某一字符串中所有包含 A 特定子串的内容用 B 特定子串替换。例如，字符串为"abc12345abc12fhgk"，A 子串为"abc"，B 子串为"ttt"，则形成的新串为"ttt12345ttt12fhgk"。
8. 有 n 个人围成一圈，顺序编号。从第一个人开始报数，（从 1 到 3 报数），凡报到 3 的人退出圈子，问最后留下的是原来第几号的那位。