1. 一维数组

#include<stdio.h>

int main()

{int a[5]={12,45,78},i;

for(i=0;i<5;i++)

printf("%3d",a[i]);

printf("\n");

for(i=4;i>=0;i--)

printf("%3d",a[i]);

return 0;

}

1. 一维数组元素排序，冒泡法，起泡法

1.方法：假设由小到大：

4,15,6,12,9

第一步：i=0:**4,15**,6,12,9，有序，不变

第二步：i=1:4,**15,6,**12,9，违背由小到大，交换

4,6,15,12,9

第三步：i=2:4,6,**15,12**,9，违背由小到大，交换

4,6,12,15,9

第四步：i=3:4,6,12,**15,9，**违背由小到大，交换

4,6,12,9,15

上述跑一趟，排好一个最大数

2.实现，基本作法：

#include<stdio.h>

int main()

{int a[5]={89,45,38,9,7},j,i,t;//由小到大

for(j=0;j<5;j++)

{

for(i=0;i<5-1;i++)

if (a[i]>a[i+1]) {t=a[i];a[i]=a[i+1];a[i+1]=t;}

for(i=0;i<5;i++)

printf("%3d",a[i]);

printf("\n");

}

return 0;

}

3.改进：

#include<stdio.h>

int main()

{int a[5]={89,45,38,9,7},j,i,t;//由小到大

for(j=0;j<5;j++)

for(i=0;i<5-1;i++)

if (a[i]>a[i+1]) {t=a[i];a[i]=a[i+1];a[i+1]=t;}

for(i=0;i<5;i++)

printf("%3d",a[i]);

printf("\n");

return 0;

}

4.实例：对七个数由大到小排序

#include<stdio.h>

int main()

{int a[7]={89,145,38,9,97,31,77},j,i,t;//由大到小

for(i=0;i<7;i++)

for(j=0;j<7-1;j++)

if (a[j]<a[j+1]){t=a[j];a[j]=a[j+1];a[j+1]=t;}

for(i=0;i<7;i++)

printf("%3d",a[i]);

printf("\n");

return 0;

}

二、查找数据：

1.简单查找：

在数组，查找指定元素，给结论

#include<stdio.h>

int main()

{int a[7]={89,145,38,9,97,31,77},i,x,di;//由大到小

scanf("%d",&x);//17

for(i=0;i<7;i++)

if (a[i]==x) break;

if (i<7) printf("找到了，在%d位置上",i+1);

else printf("没有找到");

return 0;

}

2.二分查找算法

12,31,45,**67**,52,89,98（在有序数组中查找数据）**91**

Bot=0;top=6;mid=(bot+top)/2=3

91>a[mid] :bot=mid+1,top=6不变，计算新的mid，重复

若给的数据是12：

12<a[mid]:bot=0不变，top=mid-1,计算新的mid，重复

程序：

#include<stdio.h>

int main()

{int a[7]={11,22,33,44,55,66,77},x,mid,bot,top,f=0;

scanf("%d",&x);

bot=0;top=6;

while(bot<=top)

{mid=(bot+top)/2;

if (a[mid]==x) {f=1;break;}

if (a[mid]<x) bot=mid+1;

if (a[mid]>x) top=mid-1;

}

if (f==1) printf("在%d位置上找到",mid);

else printf("没有找到");

return 0;

}

二、二维数组

1.基本结构

A[3][4]代表三行四列，并且下标均由0开始

A[0][0] a[0][1] a[0][2] a[0][3]

A[1][0] a[1][1] a[1][2] a[1][3]

A[2][0] a[2][1] a[2][2] a[2][3]

以下是初始化二维数组

#include<stdio.h>

int main()

{int a[3][4]={1,2,3,4,5,6,78,90};

//按照逐行依次赋值，不够给零

int b[3][4]={{1},{1,3},{4,5,1}}

//按照分行！依次赋值，每行不够补零

return 0;

}

2.二维数组矩阵方式输出

#include<stdio.h>

int main()

{int a[3][4]={1,2,3,4,5,6,78,90},i,j;

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)//输出每行的四个元素不换行

printf("%3d\n",a[i][j]);

printf("\n");//单独输出一个换行符

}

return 0;

}

3.找出二维数组中的最大值，以及下标

#include<stdio.h>

int main()

{int a[3][4]={1,92,3,14,5,9,178,90,31,67,87,45},i,j;

int max,col,row;

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)//输出每行的四个元素不换行

printf("%3d",a[i][j]);

printf("\n");//单独输出一个换行符

}

printf("以上是二维数组\n");

max=a[0][0];row=0;col=0;

for(i=0;i<3;i++)

for(j=0;j<4;j++)

if (a[i][j]>max) {max=a[i][j];row=i;col=j;}

printf("最大值是%d,在%d行%d列上",max,row,col);

return 0;

}

思考：可以上述结合

#include<stdio.h>

int main()

{int a[3][4]={1,92,3,814,5,9,178,90,31,67,87,45},i,j;

int max,col,row;

max=a[0][0];row=0;col=0;

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)//输出每行的四个元素不换行

{if (a[i][j]>max) {max=a[i][j];row=i;col=j;}

printf("%4d",a[i][j]);

}

printf("\n");//单独输出一个换行符

}

printf("以上是二维数组\n");

printf("最大值是%d,在%d行%d列上",max,row,col);

return 0;

}

**作业：**

1. **初始化一维数组，找出其中的最大值以及下标**
2. **初始化一维数组，由大到小排序，输出结果**
3. **初始化二维数组，以矩阵方式输出**

**交作业时间：周三上午前**