



实验5

1 (10分)

随机产生10个30~100(包括30, 100)的正整数，求它们的最大值、最小值、平均值，并显示整个数组的值和结果。

```
1  /*随机产生10个30~100(包括30, 100)的正整数，求它们的最大值、最小值、平均值，
2  并显示整个数组的值和结果。*/
3
4  #include <iostream>
5  using namespace std;
6  int main()
7  {
8      //初始化+定义数组
9      int a[10],i,max,min;
10     float ave;
11     for(i=0;i<10;i++)
12     {
13         a[i]=rand()%71+30;    //rand() 函数会返回一个介于0到RAND_MAX之间的伪
14         //随机整数
15     }
16     //显示数组内容
17     cout<<"数组a的内容如下:\n";
18     for(i=0;i<10;i++)
```

```

19     {
20         cout<<a[i]<<' ';
21     cout<<endl;
22     }
23
24 //计算各个值
25     max=min=ave=a[0];
26     for(i=0;i<10;i++)
27     {
28         ave=ave+a[i];
29         if(a[i]>max)
30             max=a[i];
31         if(a[i]<min)
32             min=a[i];
33     }
34
35 //输出结果
36     cout<<"最大值是"<<max;
37     cout<<"最小值是"<<min;
38     cout<<"平均值是"<<ave/10;
39     cout<<endl;
40
41     system("pause");
42     return 0;
43 }

```

2 (10分)

随机产生20个学生的计算机课程的成绩（0~100），按照从大到小的顺序排序，分别显示排序前和排序后的结果。

```

1  /*随机产生20个学生的计算机课程的成绩（0~100），按照从大到小的顺序排序，
2  分别显示排序前和排序后的结果。*/
3
4  #include "iostream"
5  using namespace std;
6  int main()
7  {
8  //初始化+定义数组
9      int a[20],i,j,t;
10     for(i=0;i<20;i++)
11     {

```

```

12         a[i]=rand()%101;    //rand() 函数会返回一个介于0到RAND_MAX之间的伪随
    机整数
13     }
14
15     /*
16     //显示数组内容
17         cout<<"数组a的内容如下:\n";
18         for(i=0;i<20;i++)
19         {
20             cout<<a[i]<<' ';
21             cout<<endl;
22         }
23     */
24
25     //冒泡排序
26     for(i=0;i<20;i++)
27     {
28         for(j=i+1;j<20;j++)
29         {
30             if(a[i]<a[j])
31             {
32                 t = a[i];
33                 a[i]=a[j];
34                 a[j]=t;
35             }
36         }
37     }
38
39     //输出结果
40     for(i=0;i<20;i++)
41         cout<<a[i]<<'\\t';
42     cout<<endl;
43
44     system("pause");
45     return 0;
46 }

```

3 (10分)

随机产生10个数，输入1~10之间的正整数m，使数组元素右移m位，移出的元素再从左移入。

如，假设原来的数组元素依次为：1 2 3 4 5 6 7 8 9 10，假设m为2，则右移2位后的数组元素依次为：9 10 1 2 3 4 5 6 7 8

这段代码平移有问题，不是移动的方式

```

1 //对数组进行平移
2     int t;
3     for (i=0;i<10;i++)
4     {
5         if(i<m)
6             a[i]=a[i+10-m];
7         else
8             a[i]=a[i-m];
9     }

```

修改之后:

```

1 for(j=0;j<m;j++)
2 {
3     t=s[9];
4     for(i=8;i>=0;i--)
5         s[i+1]=s[i];
6     s[0]=t;
7 }

```

```

1 /*随机产生10个数，输入1~10之间的正整数m，使数组元素右移m位，移出的元素再从左移入。
2 如，假设原来的数组元素依次为：1 2 3 4 5 6 7 8 9 10，假设m为2，则右移2位后的数
   组元素依次为：
3 9 10 1 2 3 4 5 6 7 8*/
4
5 #include "iostream"
6 using namespace std;
7 int main()
8 {
9
10 //随机产生10个数与输入m
11     int a[10],i,m;
12     cout<<"数组为:"<<"\n";
13     for(i=0;i<10;i++)
14     {
15         a[i]=rand()%11;
16         cout<<a[i]<<endl;
17     }
18     cout<<"请输入位移量"<<endl;
19     cin>>m;
20
21 //对数组进行平移

```

```

22     int j,t;
23     for(j=0;j<m;j++) //平移次数
24     {
25         t=a[9]; //t赋值最后一个
26         for(i=8;i>=0;i--) a[i+1]=a[i]; //从倒数第二个开始开始每个右移
27         a[0]=t; //第一个赋值最后一个
28     }
29 //打印数组
30     for(i=0;i<10;i++)
31         cout<<a[i]<<'\\n';
32     cout<<"输出完毕!"<<endl;
33
34     system("pause");
35     return 0;
36 }

```

4 (10分)

按由大到小的顺序输入10个int类型的数据将其存放在一个一维数组中，再输入一个整数到变量x，用二分法查找x是否是数组中的元素，若是，输出其在数组中的位置，否则输出不在数组中的提示。

这段代码是一个循环结构，具体来说是一个二分查找算法的实现。它是一个 `while` 循环，用于在有序数组 `a` 中查找元素 `x` 的位置。这里 `bot` 和 `top` 分别代表数组的下界和上界，`mid` 是中间位置的索引。

循环的逻辑是这样的：

1. 只要 `bot` 小于等于 `top`，循环就继续执行。
 2. 如果中间元素 `a[mid]` 等于要查找的元素 `x`，则 `break` 语句会跳出循环。
 3. 如果 `x` 大于中间元素 `a[mid]`，则将 `top` 更新为 `mid - 1`，缩小搜索范围到数组的左半部分。
 4. 如果 `x` 小于中间元素 `a[mid]`，则将 `bot` 更新为 `mid + 1`，缩小搜索范围到数组的右半部分。
- 这个循环会一直执行，直到找到元素 `x` 或者 `bot` 和 `top` 交叉（即 `bot` 大于 `top`），表示元素 `x` 不在数组中。

```

1  /*
2  按由大到小的顺序输入10个int类型的数据将其存放在一个一维数组中，再输入一个整数到变量x，
3  用二分法查找x是否是数组中的元素，若是，输出其在数组中的位置，否则输出不在数组中的提示。
4  */
5

```

```

6 #define N 10
7 #include "iostream"
8 using namespace std;
9
10 int main()
11 {
12
13 //输入数组和需要查找的数
14     int a[N],i;          cout << "请按递减顺序输入10个整数" << endl;
15     for (i = 0; i < N; i++)
16     {
17         cin >> a[i];
18     }
19     cout << "请输入要查找的整数" << endl;
20
21     system("pause");
22     return 0;
23
24 //二分法
25     int x,mid,bot,top;    //定义指针
26     bot=0;
27     top=N-1;
28     mid=(bot+top)/2;
29     while(bot<=top)
30     {
31         if(a[i]==a[mid])
32             break;
33         else if(x > a[mid])
34             top=mid - 1;
35         else
36             bot = mid + 1;
37     }
38     if (bot > top)
39         cout << x << "不在数组中" << endl;
40     else
41         cout << x << "是数组中的第" <<mid+1<<"个元素"<< endl;
42
43     system("pause");
44     return 0;
45 }

```