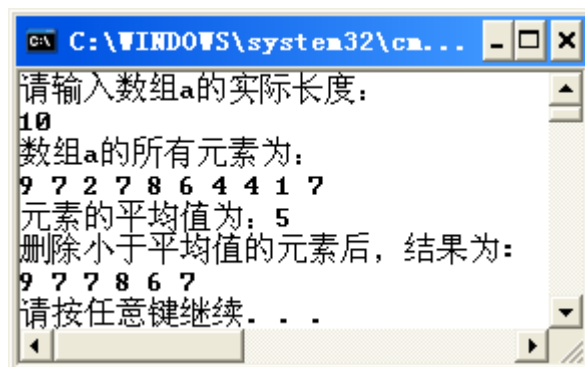


```
#include<iostream>
#include<iomanip>
using namespace std;
void func(int a[],int& n)    //数组的长度用引用变量做形参，引用变量就是在其前面加&，那么对形参值的改变会影响到实参的值！
{
    int i,j,pos=1; //将数组a的第1个元素看成默认的结果集，则pos指示结果集的下一个写入位置
    for(i=1;i<n;i++)    //从a的第2个元素依次处理
    {
        for(j=0;j<pos;j++)
            if(a[j]==a[i]) //如果待处理的元素已经在结果集中，则不将其写入结果集
                break;    //结束本轮处理
        if(j>=pos)    //或者j==pos
        {
            a[pos]=a[i];
            pos++;
        } //如果当前处理的元素不在结果集中，则将其写入结果集由pos指示的位置
    }
    n=pos;    //我们讲的去重算法，并未真正从数组中删除重复元素，只是把非重复的元素放在了数组的前部，
              //所以，如果不改变长度值n，会把所有元素都输出
              //将去重后的结果集长度pos赋给n，因n为引用变量，故它的改变会返回给实参
}
void main()
{
    const int N=20;
    int a[N];
    int i,n;
    cout<<"请输入数组长度";
    cin>>n;
    cout<<"请输入该数组"<<endl;
    for(i=0;i<n;i++)
```

```

        cin>>a[i];
    func(a,n);    //用引用变量做形参,函数调用时,对应的实参只需给出值即可,
    所有调用形式还是: 函数名(数组名, 数组长度)
    //调用完成后, 数组a中的元素进行了调整(前一部分保留去重后的结果集,
    后一部分为剩余元素), 同时n改变为结果集的长度!
    for(i=0;i<n;i++)
        cout<<setw(3)<<a[i];
}

```



```

#include<iostream>
#include <ctime>
#include<cstdlib>
using namespace std;
void main()
{
    void func(int a[],int& n,int ave);    //数组的长度为引用变量,即在删除过程
    中, a的实际长度在变量, 希望形参能返回变化的值, 故设置为引用变量
    const int N=20;
    int a[N],i,m,sum=0,ave;

    //输入数组实际长度及所有元素
    cout<<"请输入数组a的实际长度: "<<endl;
    cin>>m;
    cout<<"数组a的所有元素为: "<<endl;
    srand(time(NULL));
    for(i=0;i<m;i++)
    {
        a[i]=1+rand()%10;
        sum=sum+a[i];
        cout<<a[i]<<" ";
    }
    ave=sum/m;
    cout<<endl<<"元素的平均值为: "<<ave<<endl;
}

```

```

func(a,m,ave);

cout<<"删除小于平均值的元素后，结果为:"<<endl;
for(i=0;i<m;i++)
    cout<<a[i]<<" ";
cout<<endl;
}
void func(int a[],int& n,int ave)    //数组的长度用引用变量做形参，引用变量就是
    在其前面加&，那么对形参值的改变会影响到实参的值！
{
    int i,j;
    for(i=0;i<n;i++)    //处理a中所有元素
    {
        if(a[i]<ave)
        {
            for(j=i;j<n;j++)
                a[j]=a[j+1];    //删除小于平均值的元素
            n--;    //数组a的实际长度减
            i--;    //回溯下标，以保证不漏掉元素
        }
    }
}

```