

### 【问题描述】

从标准输入中输入一组互不相同的整数（个数不超过 100）及排序方式，按照从小到大排序，输出按某种算法排序的结果及元素的比较次数。

说明：排序方式为一个 1~5 的整数，分别表示：

- 1：选择排序，比较次数是指选择未排序部分的最小元素时的比较次数。
- 2：冒泡排序，比较次数是指相邻元素的比较次数，若某趟排序中没有进行数据交换，就认为排序结束。
- 3：堆排序，比较次数是指根元素调整过程中根元素与子树根结点的比较次数，即下面算法中红色语句的执行次数：

```
void adjust(int k[ ],int i,int n)
```

```
{
    int j,temp;
    temp=k[i];
    j=2*i+1;
    while(j<n){
        if(j<n-1 && k[j]<k[j+1])
            j++;
        if(temp>=k[j])
            break;
        k[(j-1)/2]=k[j];
        j=2*j+1;
    }
    k[(j-1)/2]=temp;
}
```

- 4：二路归并排序，比较次数是指两组有序数据合并成一组时的比较次数，即下面算法中红色语句的执行次数（注意：调用 merge 时，要使用上课讲的递归算法）：

```
void merge(int x[ ],int tmp[ ],int left,int leftend,int rightend)
```

```
{
    int i=left, j=leftend+1, q=left;
    while(i<=leftend && j<=rightend)
    {
        if(x[i]<=x[j])
            tmp[q++]=x[i++];
        else
            tmp[q++]=x[j++];
    }
    while(i<=leftend)
        tmp[q++]=x[i++];
    while(j<=rightend)
        tmp[q++]=x[j++];
    for(i=left; i<=rightend; i++)
        x[i]=tmp[i];
}
```

5: 快速排序，比较次数是指分界元素与其它元素的比较次数，即下面算法中红色语句的执行次数：

```
void quickSort(int k[ ],int left,int right)
{
    int i, last;
    if(left<right){
        last=left;
        for(i=left+1;i<=right;i++)
            if(k[i]<k[left])
                swap(&k[++last],&k[i]);
        swap(&k[left],&k[last]);
        quickSort(k,left,last-1);
        quickSort(k,last+1,right);
    }
}
```

#### 【输入形式】

首先在屏幕上输入 2 个整数，分别表示待排序的整数个数及排序方式，然后在下一行依次输入待排序的整数。各整数之间都以一个空格分隔。

#### 【输出形式】

先在一行上输出排序结果，各整数间以一个空格分隔。然后在下一行上输出排序过程中的元素比较次数。

#### 【样例 1 输入】

```
20 1
38 356 98 -102 126 46 65 -9 100 0 21 2 90 8 18 12 78 16 189 23
```

#### 【样例 1 输出】

```
-102 -9 0 2 8 12 16 18 21 23 38 46 65 78 90 98 100 126 189 356
190
```

#### 【样例 1 说明】

输入了 20 个整数数据，要求按照选择排序算法对输入的数据进行从小到大排序，输出排序结果，排序过程中元素的比较次数为 190 次。

#### 【其它样例说明】

若输入的待排序数据与样例 1 完全相同，要求的排序算法不同，则输出的排序结果与样例 1 完全一样，但比较次数不同，为了方便说明，下面左侧为排序方式，右侧为对应的比较次数：

2	162
3	58
4	66
5	75

#### 【评分标准】

该题要求按照指定算法对输入的数据进行排序，提交程序名为 sort.c。