

## 问题 A: 输出数组中能被其他元素整除的元素

时间限制: 1 Sec 内存限制: 128 MB

提交: 404 解决: 335

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

### 题目描述

输入20个整数，输出其中能被数组中其它元素整除的那些数组元素。

### 输入

### 输出

### 样例输入

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

### 样例输出

4  
6  
8  
9  
10  
12  
14  
15  
16  
18  
20  
21

```
#include<stdio.h>

int main()
{
    int a[100],b[100],sum=0;
    for(int i=1;i<=20;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=1;i<=20;i++)
    {
        for(int j=1;j<=20;j++)
        {
            if(i!=j&&a[i]%a[j]==0)
            {
                b[++sum]=a[i];
                break;
            }
        }
    }
}
```

```
    }  
    }  
    }  
    for(int i=1;i<=sum;i++)  
    {  
        printf("%d\n",b[i]);  
    }  
    return 0;  
}
```

## 问题 B: 在有序数列中插入元素

时间限制: 1 Sec 内存限制: 128 MB

提交: 261 解决: 189

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

### 题目描述

已有一个已排好的9个元素的数组，今输入一个数要求按原来排序的规律将它插入数组中。

### 输入

第一行，原始数列。第二行，需要插入的数字。

### 输出

排序后的数列

### 样例输入

```
1 7 8 17 23 24 59 62 101  
50
```

### 样例输出

```
1  
7  
8  
17  
23  
24  
50  
59  
62  
101
```

```
#include<stdio.h>
int main()
{
    int a[10], i, j, num, t;
    for (i=0; i<9; i++)
        scanf("%d", &a[i]);
    scanf("%d", &num);
    if (num>a[8])
        a[9]=num;
    else
    {
        for (i=0; i<9; i++)
        {
            if (num<a[i])
            {
                for (j=8; j>=i; j--)
                    a[j+1]=a[j];
                a[i]=num;
                break;
            }
        }
    }
    for (i=0; i<10; i++)
        printf("%d\n", a[i]);
    return 0;
}
```

## 问题 C: 选择排序 (数组)

时间限制: 1 Sec 内存限制: 128 MB

提交: 1505 解决: 732

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

### 题目描述

输入n个数字，使用选择排序对齐进行升序排序。

### 输入

测试次数T

每组测试数据格式为：数字个数n，后跟n个数字，假设n<20

### 输出

对每组测试数据，输出选择排序的每趟排序结果

### 样例输入

```
2
5 10 4 43 1 2
10 20 123 12 2 434 54 2 45 23 -10
```

### 样例输出

```
1 4 43 10 2
1 2 43 10 4
1 2 4 10 43
1 2 4 10 43
-10 123 12 2 434 54 2 45 23 20
-10 2 12 123 434 54 2 45 23 20
-10 2 2 123 434 54 12 45 23 20
-10 2 2 12 434 54 123 45 23 20
-10 2 2 12 20 54 123 45 23 434
-10 2 2 12 20 23 123 45 54 434
-10 2 2 12 20 23 45 123 54 434
-10 2 2 12 20 23 45 54 123 434
-10 2 2 12 20 23 45 54 123 434
```

```
#include<stdio.h>

int main()
{
    int t,n,a[10000],minn,flag;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);
```

```
    for(int i=1;i<=n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=1;i<n;i++)
    {
        minn=a[i];
        flag=0;
        for(int j=i+1;j<=n;j++)
        {
            if(a[j]<minn)
            {
                minn=a[j];
                flag=j;
            }
        }
        if(flag!=0)
        {
            int qwe;
            qwe=a[i];
            a[i]=a[flag];
            a[flag]=qwe;
        }
        for(int j=1;j<n;j++)
        {
            printf("%d ",a[j]);
        }
        printf("%d\n",a[n]);
    }
}

return 0;
```

```
}
```

## 问题 D: 学生成绩管理系统 - I

时间限制: 1 Sec 内存限制: 128 MB

提交: 1889 解决: 675

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

### 题目描述

现有一学生管理系统，包含N个学生姓名与学生成绩，你需要回答Q个操作，这些询问包括求平均值，求最大值，求最小值，改变某个同学的成绩

### 输入

测试数据有多组

输入的第一行输入一个整数T，代表测试组数

每组测试数据第一行有两个整数N, M ( $1 \leq N \leq 100$ ,  $1 \leq M \leq 100$ )

接下来N行输入N个学生的姓名和成绩，保证每个同学的姓名各不相同且长度不超过20，成绩都是非负整数

接下来M行输入M个操作，这些操作的格式包括：

**QUERY\_AVERAGE**: 求平均值，输出保留1位小数

**QUERY\_MAX**: 求最大值

**QUERY\_MIN**: 求最小值

**CHANGE index score**: 将存在于名单中下标为index（下标从1开始）的同学的成绩修改为score，输入保证合法

### 输出

对于每组测试样例

对于其中的每个询问操作各输出一行

每组样例之间输出一行空行

### 样例输入

```
2
3 7
DalaoLin 100
XuezhaChen 59
ChiguaWang 88
QUERY_AVERAGE
QUERY_MAX
QUERY_MIN
CHANGE 1 99
QUERY_MIN
QUERY_MAX
QUERY_AVERAGE
1 1
DalaoLin 100
QUERY_AVERAGE
```

### 样例输出

```
82.3
100
59
59
99
82.0
100.0
```

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

```

int main()
{
    int T, M, N, min, max, x, y;
    char name[100][20], op[100];
    int s[100], i, j, k, t;
    double average, sum;
    scanf("%d", &T);
    for (i=1; i<=T; i++)
    {
        scanf("%d %d", &N, &M);
        for (j=0; j<N; j++)
            scanf("%s %d", name, &s[j]);
        for (k=1; k<=M; k++)
        {
            max=s[0];
            min=s[0];
            sum=0;
            average=0;
            scanf("%s", op);
            if ((strcmp(op, "QUERY_AVERAGE"))==0)
            {
                for (j=0; j<N; j++)
                    sum=sum+s[j];
                average=sum/N;
                printf("%.1lf\n", average);
            }
            else if ((strcmp(op, "QUERY_MAX"))==0)
            {
                for (j=0; j<N; j++)
                {
                    if (max<s[j])
                        max=s[j];
                }
                printf("%d\n", max);
            }
            else if ((strcmp(op, "QUERY_MIN"))==0)
            {
                for (j=0; j<N; j++)
                {
                    if (min>s[j])
                        min=s[j];
                }
                printf("%d\n", min);
            }
        }
    }
}

```

```

        else
        {
            scanf("%d %d",&x,&y);
            s[x-1]=y;
        }
    }
    printf("\n");
}
return 0;
}

```

### 问题 E: 约瑟夫环(数组)

时间限制: 1 Sec 内存限制: 128 MB

提交: 1202 解决: 714

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

#### 题目描述

有n个人围成一个圈，从第一个人开始顺序报号1, 2, 3。凡是报到3退出圈子中的人原来的序号。要求打印出退出人的序号。以及找出最后留在圈子中的人原来的序号。

#### 输入

输入n

#### 输出

退出人的序号，并找出最后留在圈子里的人原来的序号

#### 样例输入

13

#### 样例输出

3 6 9 12 2 7 11 4 10 5 1 8  
13

```

#include<stdio.h>
int main()
{
    int i,number,count=1,array[100],k=0,n;
    scanf("%d",&n);
    number=n;
    for(i=0;i<n;i++)
        array[i]=i+1;
    while(number>1)
    {
        if(array[k]!=0)
        {
            if(count<3)
            {
                count++;
            }
        }
    }
}

```



```
        k=(k+1)%n;
    }
    else
    {
        printf("%d ", array[k]);
        count=1;
        array[k]=0;
        number--;
        k=(k+1)%n;
    }
}
else
    k=(k+1)%n;
}
printf("\n");
for(i=0;i<n;i++)
{
    if(array[i]>0)
        printf("%d\n", array[i]);
}
return 0;
}
```

## 问题 F: 求矩阵每列的最大值与和

时间限制: 1 Sec 内存限制: 128 MB

提交: 1650 解决: 1171

[\[提交\]](#)[\[状态\]](#)[\[讨论版\]](#)

### 题目描述

定义一个二维数组（不超过10\*10），输入方阵的大小n，然后输入数组的数据，求该矩阵每列的最大值与和

### 输入

测试数据的组数t

第一个矩阵的大小n

第一个矩阵的元素

第二个矩阵的大小n

第二个矩阵的元素

.....

### 输出

第一个矩阵的每一列最大值、和

第二个矩阵的每一列最大值、和

### 样例输入

```
2
3
12 25 3 41 5 66 7 82 91
4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

### 样例输出

```
41 60
82 112
91 160
13 28
14 32
15 36
16 40
```

```
#include<stdio.h>

int main()
{
    int t,n,a[100][100],maxx,sum;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);
```

```
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                scanf("%d",&a[i][j]);
            }
        }
        for(int i=1;i<=n;i++)
        {
            sum=0;
            maxx=0;
            for(int j=1;j<=n;j++)
            {
                sum+=a[j][i];
                if(a[j][i]>maxx)
                {
                    maxx=a[j][i];
                }
            }
            printf("%d %d\n",maxx,sum);
        }
    }
    return 0;
}
```