

# 天勤论坛 (www.csbiji.com)

第一题: xxx 定律: 数  $n$ , 如果是偶数, 把  $n$  砍掉一半; 如果奇数, 把  $3n+1$  砍掉一半, 直到数变为 1 为止。测试包含多个用例, 当数为 0 时结束。输出需要经过几步将  $n$  变到 1;

Input:

3

1

0

Output:

5

0

```
1. #include <stdio.h>
2. int main()
3. {
4.     int n,s;
5.     while (scanf("%d",&n),n)
6.     {
7.         s=0;
8.         while (n!=1)
9.         {
10.             if (n%2==0) n/=2;
11.             else n=(3*n+1)/2;
12.             s++;
13.         }
14.         printf("%d\n", s);
15.     }
16.     system("pause");
17.     return 0;
18. }
```

第二题: ZOJ, 读入一个字符串, 字符串中包含 ZOJ 三个字符, 个数不一定相等, 按 ZOJ 的顺序输出, 当某个字符用完时, 剩下的仍然安装 ZOJ 的顺序输出。当读入的字符串为 E 时, 结束。

Input:

ZZOOOJJJ

ZZZZO0000OJJJ

E

Output:

ZOJZOJOJ

ZOJZOJZOJZOO

```
1. #include <stdio.h>
2. #include <string.h>
3. int main()
4. {
5.     char s[1000];
6.     int i,Zn,On,Jn;
7.     while (gets(s) && strcmp(s,"E"))
8.     {
9.         Zn=On=Jn=0;
10.        for (i=0; i<strlen(s); i++)
11.        {
12.            if (s=='Z')    Zn++;
13.            else if (s=='O')    On++;
14.            else    Jn++;
15.        }
16.        while (Zn)
17.        {
18.            printf("Z");
19.            Zn--;
20.            if (On)
21.            {
22.                printf("O");
23.                On--;
24.            }
25.            if (Jn)
26.            {
27.                printf("J");
```

```
28.          Jn--;
29.          }
30.      }
31.      while (On)
32.      {
33.          printf("0");
34.          On--;
35.          if (Jn)
36.          {
37.              printf("J");
38.              Jn--;
39.          }
40.      }
41.      while (Jn)
42.      {
43.          printf("J");
44.          Jn--;
45.      }
46.      printf("\n");
47.  }
48.  system("pause");
49.  return 0;
50. }
```

第三题：继续 xxx 定律（还是第一题的定律），当 n 为 3 时，我们在验证 xxx 定律的过程中会得到一个序列，3，5，8，4，2，1，将 3 称为关键数，5，8，4，2 称为覆盖数。现在输入 n 个数字，根据关键数与覆盖数的理论，我们只需要验证其中部分数就可以确定所有数满足 xxx 定律，最后输出的 n 个数中的关键数。

Input: 第一行输入 n，第二行输入 n 个数，n 为 0 时结束

3

3 8 4

0

Output: 有多个关键数时逆序输出

```
1. #include <stdio.h>
2. #define N 1000000
3. int tmp[N+1],Hash[N+1];
4. int main()
5. {
6.     int n,i,s;
7.     while (scanf("%d",&n),n)
8.     {
9.         memset(Hash,0,sizeof(Hash));
10.        for (i=0; i<n; i++)
11.        {
12.            scanf("%d",&tmp);
13.            s=tmp;
14.            do {
15.                if (s%2==0) s/=2;
16.                else s=(3*s+1)/2;
17.                Hash[s]=1;
18.            } while (s!=1);
19.        }
20.        for (i=n-1; i>=0; i--)
21.        {
22.            //printf("tmp[%d]=%d\n", i,tmp);
23.            if (!Hash[tmp]) printf("%d\n", tmp);
24.        }
25.    }
26.    system("pause");
27.    return 0;
28. }
```

# 天勤论坛 (www.csbiji.com)

第四题：寻找大富翁；输入  $n$  ( $0 < n \leq 100000$ )， $m$  ( $0 < m \leq 10$ )， $n$  为小镇上的人数， $m$  为需要找出的大富翁数。输入：  
 $n, m$  接下来一行输入小镇  $n$  个人的财富值，输出：前  $m$  个大富翁的财产数， $n$  为 0 时结束

Input:

3 1  
2 5 -1  
0 0

Output:

5

```
1. #include <stdio.h>
2. int n,m,w[100001],max[20];
3. int main()
4. {
5.     int i,j,tmp;
6.     while (scanf("%d%d",&n,&m),n)
7.     {
8.         for (i=0; i<n; i++) scanf("%d",&w);
9.         //max=w[0];
10.        for (i=0; i<m; i++)
11.        {
12.            //max=w;
13.            for (j=i; j<n; j++)
14.                if (w[j]>w)
15.                {
16.                    tmp=w;
17.                    w=w[j];
18.                    w[j]=tmp;
19.                }
20.            max=w;
21.        }
22.        for (i=0; i<m-1; i++) printf("%d ",max);
23.        printf("%d\n",max);
```

```
24.     }  
25.     system("pause");  
26.     return 0;  
27. }
```

**注意：这一道不能 sort 直接排序，不然会超时。可以用堆排序或者选择排序都可以。**

第五题: 找出直系亲属。如果 A, B 是 C 的父母亲, 则 A, B 是 C 的 parent, C 是 A, B 的 child, 如果 A, B 是 C 的 (外) 祖父, 祖母, 则 A, B 是 C 的 grandparent, C 是 A, B 的 grandchild, 如果 A, B 是 C 的 (外) 曾祖父, 曾祖母, 则 A, B 是 C 的 great-grandparent, C 是 A, B 的 great-grandchild, 之后再多一辈, 则在关系上加一个 great-。

输入: n ( $0 \leq n \leq 26$ ) 表示 n 个亲属关系, 形式为 ABC, 表示 A 的父母亲分别是 B, C, 如果 A 的父母亲信息不全, 则用-代替, 例如 A-C. m (。。。) 代表测试用例数, 形式 AB。输出: AB 的关系, 如 A 是 B 的直系亲属, 按上述要求输出关系, 如果 A, B 没有关系, 输出-。当 n 为 0 时结束。

Input:

```
3 2  
ABC  
CDE  
EFG  
FA  
BE  
0 0
```

Output:

```
great-grandparent  
-
```

```
1. #include <stdio.h>  
2. typedef struct _Node  
3. {  
4.     int p1,p2,child;
```

```
5. } Node;

6. Node parent[26];

7. int n,m;

8. int main()

9. {

10.     char a,b,c;

11.     int i,j,t,flag,s;

12.     while (scanf("%d%d%c",&n,&m),n)

13.     {

14.         memset(parent,-1,sizeof(parent));

15.         for (i=0; i<n; i++)

16.         {

17.             scanf("%c%c%c%c",&a,&b,&c);

18.             if (b>='A'&&b<='Z')

19.             {

20.                 parent[a-'A'].p1=b-'A';

21.                 parent[b-'A'].child=a-'A';

22.             }

23.             if (c>='A'&&c<='Z')

24.             {

25.                 parent[a-'A'].p2=c-'A';

26.                 parent[c-'A'].child=a-'A';

27.             }

28.         }

29.         while (m--)

30.         {

31.             scanf("%c%c%c",&a,&b);

32.             t=a-'A';

33.             flag=0;

34.             s=0;

35.             while (t>=0)

36.             {
```

```
37.          //printf("%d schild is %d\n", t,parent[t].child);
38.          t=parent[t].child;
39.          s++;
40.          if (t==b-'A')
41.          {
42.              flag=1;
43.              break;
44.          }
45.      }
46.      if (flag)
47.      {
48.          if (s==1)  printf("parent\n");
49.          else if (s==2)  printf("grandparent\n");
50.          else
51.          {
52.              while (s-->2)
53.              {
54.                  printf("great-");
55.              }
56.              printf("grandparent\n");
57.          }
58.      }
59.      else  printf("-\n");
60.  }
61.  }
62.  system("pause");
63.  return 0;
64. }
```