

# ABC 089 解説

writer: namonakiacc

平成 30 年 3 月 4 日

For international readers: English Editorial starts at page 4.

## A: Grouping 2

3 人以上のグループは最大で  $\lfloor N/3 \rfloor$  個作れます。

よって、 $\lfloor N/3 \rfloor$  を出力するとよいです。

```
#include <cstdio>
int N;
int main()
{
    scanf("%d",&N);
    printf("%d\n",N/3);
}
```

## B: Hina Arare

袋には必ず桃色、白色、緑色のひなあられが入っていることが保証されるので、3 種類か 4 種類かの判別は黄色のひなあられの有無で判別できます。

よって、黄色のひなあられがあれば 'Four'、なければ 'Three' と出力するとよいです。

```
#include <cstdio>
int N;
char c;
int main()
{
    scanf("%d",&N);
    bool flag=false;
    for(int i=0;i<N;i++)
    {
        scanf("%c",&c);
        if(c=='Y')flag=true;
    }
    if(flag)puts("Four");
    else puts("Three");
}
```

## C: March

‘M’ から始まる名前の人が  $m$  人、‘A’ から始まる名前の人が  $a$  人、‘R’ から始まる名前の人が  $r$  人、‘C’ から始まる名前の人が  $c$  人、‘H’ から始まる名前の人が  $h$  人いるとします。

例えば、「‘M’ から始まる名前の人」と「‘A’ から始まる名前の人」と「‘R’ から始まる名前の人」の 3 人を選ぶ方法は、 $m \times a \times r$  通りです。

3 人を選ぶとき、3 人の名前の頭文字の組み合わせは高々 10 通りとなります。

よって、 $m, a, r, c, h$  を求めた後、10 通りの組み合わせ全てを試せばよいです。

```
#include <cstdio>
#include <iostream>
using namespace std;
typedef long long ll;

string s;
int N;
ll m,a,r,c,h;
ll D[5];
int P[10]={0,0,0,0,0,0,1,1,1,2};
int Q[10]={1,1,1,2,2,3,2,2,3,3};
int R[10]={2,3,4,3,4,4,3,4,4,4};
int main()
{
    scanf("%d",&N);
    for(int i=0;i<N;i++)
    {
        cin>>s;
        if(s[0]=='M')m++;
        if(s[0]=='A')a++;
        if(s[0]=='R')r++;
        if(s[0]=='C')c++;
        if(s[0]=='H')h++;
    }
    D[0]=m,D[1]=a,D[2]=r,D[3]=c,D[4]=h;
    ll res=0;
    for(int d=0;d<10;d++)
        res+=D[P[d]]*D[Q[d]]*D[R[d]];
    printf("%lld\n",res);
}
```

## D: Practical Skill Test

初めに、1 以上  $H \times W$  以下の全ての整数に対して、その整数がどのマスに書かれているかを記録しておきます。

以下、整数  $i$  はマス  $(px_i, py_i)$  に書かれているものとします。

次に、 $D$  が全てのクエリにおいて共通の値となっているので、 $D$  を利用することを考えます。

$R_i - L_i$  が  $D$  の倍数であることが保証されているため、 $D$  で割った余りを用いて累積和の方式を用いることができます。

- $d_i = 0$  ( $1 \leq i \leq D$ )
- $d_i = d_{i-D} + |px_i - px_{i-D}| + |py_i - py_{i-D}|$  ( $D + 1 \leq i$ )

とすると、実技試験において駒を  $L$  から  $R$  まで移動するのに消費される魔力は  $d_R - d_L$  となります。

```
#include <cstdio>
#define abs(x) ((x>0)?x:(-(x)))
int H,W,D,A;
int Q,L,R;
int px[90001],py[90001];
int d[90001];
int main()
{
    scanf("%d%d%d",&H,&W,&D);
    for(int i=0;i<H;i++)
    {
        for(int j=0;j<W;j++)
        {
            scanf("%d",&A);
            px[A]=i,py[A]=j;
        }
    }
    for(int i=D+1;i<=H*W;i++)
        d[i]=d[i-D]+abs(px[i]-px[i-D])+abs(py[i]-py[i-D]);
    scanf("%d",&Q);
    while(Q--)
    {
        scanf("%d%d",&L,&R);
        printf("%d\n",d[R]-d[L]);
    }
}
```

## A: Grouping 2

The maximum number of groups consisting of three or more students is  $\lfloor N/3 \rfloor$ .

So, you can get 100 points by print  $\lfloor N/3 \rfloor$ .

```
#include <stdio>
int N;
int main()
{
    scanf ("%d",&N);
    printf ("%d\n",N/3);
}
```

## B: Hina Arare

It is guaranteed that there are pink,white,green arare,so it is possible to distinguish the number of colors by the existence of yellow arare.

So, you can get 200 points by print ‘Four’ when yellow arare exists, ‘Three’ otherwise.

```
#include <stdio>
int N;
char c;
int main()
{
    scanf ("%d",&N);
    bool flag=false;
    for(int i=0;i<N;i++)
    {
        scanf ("%c",&c);
        if(c=='Y')flag=true;
    }
    if(flag)puts("Four");
    else puts("Three");
}
```

## C: March

Considering as  $m$  means the number of people the name begins with 'M',  $a$  - 'A',  $r$  - 'R',  $c$  - 'C',  $h$  - 'H'.

For example, the way to choose people the name begins with 'M', 'A', 'R' is  $m \times a \times r$ .

When choosing 3 people, there are no more than 10 patterns of the first character of those names.

So, it is OK to calculate  $m, a, r, c, h$  and try all 10 patterns.

```
#include <cstdio>
#include <iostream>
using namespace std;
typedef long long ll;

string s;
int N;
ll m, a, r, c, h;
ll D[5];
int P[10]={0,0,0,0,0,0,1,1,1,2};
int Q[10]={1,1,1,2,2,3,2,2,3,3};
int R[10]={2,3,4,3,4,4,3,4,4,4};
int main()
{
    scanf("%d",&N);
    for(int i=0;i<N;i++)
    {
        cin>>s;
        if(s[0]=='M')m++;
        if(s[0]=='A')a++;
        if(s[0]=='R')r++;
        if(s[0]=='C')c++;
        if(s[0]=='H')h++;
    }
    D[0]=m,D[1]=a,D[2]=r,D[3]=c,D[4]=h;
    ll res=0;
    for(int d=0;d<10;d++)
        res+=D[P[d]]*D[Q[d]]*D[R[d]];
    printf("%lld\n",res);
}
```

## D: Practical Skill Test

First, memorize where the number is written, in all number in 1 to  $H \times W$ .

Consider as number  $i$  is written in  $(px_i, py_i)$ .

Next,  $D$  is common in every query, so thinking about the way to use  $D$ .

$R_i - L_i$  can be divided by  $D$ , so it is OK to use cumulative sum.

Consider as

-  $d_i = 0$  ( $1 \leq i \leq D$ )

-  $d_i = d_{i-D} + |px_i - px_{i-D}| + |py_i - py_{i-D}|$  ( $D + 1 \leq i$ )

, the sum of magic points consumed to move a piece from  $L$  to  $R$  in test is  $d_R - d_L$ .

```
#include <cstdio>
#define abs(x) ((x>0)?x:(-(x)))
int H,W,D,A;
int Q,L,R;
int px[90001],py[90001];
int d[90001];
int main()
{
    scanf("%d%d%d",&H,&W,&D);
    for(int i=0;i<H;i++)
    {
        for(int j=0;j<W;j++)
        {
            scanf("%d",&A);
            px[A]=i,py[A]=j;
        }
    }
    for(int i=D+1;i<=H*W;i++)
        d[i]=d[i-D]+abs(px[i]-px[i-D])+abs(py[i]-py[i-D]);
    scanf("%d",&Q);
    while(Q--)
    {
        scanf("%d%d",&L,&R);
        printf("%d\n",d[R]-d[L]);
    }
}
```