

ABC 079 解説

writer: namonakiacc

平成 29 年 11 月 18 日

For international readers: English Editorial starts at page 3.

A: Good Integer

N は 4 桁の整数なので、3 つ以上の同じ数字が連続して並ぶ時、前の 3 つの数字か、後の 3 つの数字の少なくともどちらかは必ず等しくなります。

よって、この 2 つの条件の少なくともどちらかを満たしたときに 'Yes'、そうでないときに 'No' を出力するとよいです。

```
#include <cstdio>
char s[5];
int main()
{
    scanf("%s",&s);
    if(s[0]==s[1]&& s[1]==s[2]) puts("Yes");
    else if(s[1]==s[2]&& s[2]==s[3]) puts("Yes");
    else puts("No");
}
```

B: Lucas Number

リュカ数の定義 ($L_0 = 2, L_1 = 1, L_i = L_{i-1} + L_{i-2} (i \geq 2)$) に従って計算すると求まります。ただし、答えが 32 bit に収まらない場合があるので注意してください。

```
#include <cstdio>
typedef long long ll;
int N;
ll L[87];
int main()
{
    scanf("%d",&N);
    L[0]=2LL,L[1]=1LL;
    for(int i=2;i<=N;i++)L[i]=L[i-2]+L[i-1];
    printf("%lld\n",L[N]);
}
```

C: Train Ticket

op1,op2,op3 の決め方は高々 8 通りであるため、これらを全て試して当てはまるものがあれば 1 つ出力するとよいです。

```
#include <stdio>
char s[5];
int main()
{
    scanf("%s",&s);
    int a=s[0]-'0',b=s[1]-'0',c=s[2]-'0',d=s[3]-'0';
    if(a-b-c-d==7)printf("%d-%d-%d-%d=7\n",a,b,c,d);
    else if(a-b-c+d==7)printf("%d-%d-%d+%d=7\n",a,b,c,d);
    else if(a-b+c-d==7)printf("%d-%d+%d-%d=7\n",a,b,c,d);
    else if(a-b+c+d==7)printf("%d-%d+%d+%d=7\n",a,b,c,d);
    else if(a+b-c-d==7)printf("%d+%d-%d-%d=7\n",a,b,c,d);
    else if(a+b-c+d==7)printf("%d+%d-%d+%d=7\n",a,b,c,d);
    else if(a+b+c-d==7)printf("%d+%d+%d-%d=7\n",a,b,c,d);
    else if(a+b+c+d==7)printf("%d+%d+%d+%d=7\n",a,b,c,d);
}
```

D: Wall

初めに、すべての数字において、1 に変えるのにかかるコストを計算しておきます。

これには、ワーシャルフロイド法などが使えるでしょう。

その後、全てのマスにおいて、数字が書かれている場合の 1 への変更コストの和を求めるとよいです。

```
#include <stdio>
#define rep(i,n) for(int (i)=0;(i)<(n);(i)++)
int c[10][10];
int H,W,A,res;
int main()
{
    scanf("%d%d",&H,&W);
    rep(i,10)rep(j,10)scanf("%d",&c[i][j]);
    rep(k,10)rep(i,10)rep(j,10)
        if(c[i][j]>c[i][k]+c[k][j])c[i][j]=c[i][k]+c[k][j];
    rep(i,H)rep(j,W)
    {
        scanf("%d",&A);
        if(A>=0)res+=c[A][1];
    }
    printf("%d\n",res);
}
```

A: Good Integer

N has 4 digits, so first 3 or last 3 digits are same when N is **good**.

So, you can get 100 points by output 'Yes' when at least one of these conditions are met, 'No' when they didn't.

```
#include <stdio>
char s[5];
int main()
{
    scanf("%s",&s);
    if(s[0]==s[1]&& s[1]==s[2]) puts("Yes");
    else if(s[1]==s[2]&& s[2]==s[3]) puts("Yes");
    else puts("No");
}
```

B: Lucas Number

You can solve it by calculate with the definition of Lucas Number($L_0 = 2$, $L_1 = 1$, $L_i = L_{i-1} + L_{i-2}(i \geq 2)$).

Pay attention to the case when the answer can't be expressed in 32 bit.

```
#include <stdio>
typedef long long ll;
int N;
ll L[87];
int main()
{
    scanf("%d",&N);
    L[0]=2LL,L[1]=1LL;
    for(int i=2;i<=N;i++)L[i]=L[i-2]+L[i-1];
    printf("%lld\n",L[N]);
}
```

C: Train Ticket

There are no more than 8 pattern to decide $op1,op2,op3$,so you can solve it by try all patterns and output one case which is answer.

```
#include <stdio>
char s[5];
int main()
{
    scanf("%s",&s);
    int a=s[0]-'0',b=s[1]-'0',c=s[2]-'0',d=s[3]-'0';
    if(a-b-c-d==7) printf("%d-%d-%d-%d=7\n",a,b,c,d);
    else if(a-b-c+d==7) printf("%d-%d-%d+%d=7\n",a,b,c,d);
    else if(a-b+c-d==7) printf("%d-%d+%d-%d=7\n",a,b,c,d);
    else if(a-b+c+d==7) printf("%d-%d+%d+%d=7\n",a,b,c,d);
    else if(a+b-c-d==7) printf("%d+%d-%d-%d=7\n",a,b,c,d);
    else if(a+b-c+d==7) printf("%d+%d-%d+%d=7\n",a,b,c,d);
    else if(a+b+c-d==7) printf("%d+%d+%d-%d=7\n",a,b,c,d);
    else if(a+b+c+d==7) printf("%d+%d+%d+%d=7\n",a,b,c,d);
}
```

D: Wall

First, calculate cost to turn digits to 1 ,in 0,2,3,...,9.

You can do it,for example,by Warshall Floyd algorithm.

Then, you can solve it by calculate the sum of the cost to turn digits to 1,in every square which contain a digit.

```
#include <stdio>
#define rep(i,n) for(int (i)=0;(i)<(n);(i)++)
int c[10][10];
int H,W,A,res;
int main()
{
    scanf("%d%d",&H,&W);
    rep(i,10) rep(j,10) scanf("%d",&c[i][j]);
    rep(k,10) rep(i,10) rep(j,10)
        if(c[i][j]>c[i][k]+c[k][j]) c[i][j]=c[i][k]+c[k][j];
    rep(i,H) rep(j,W)
    {
        scanf("%d",&A);
        if(A>=0) res+=c[A][1];
    }
    printf("%d\n",res);
}
```