OLIMPIADA NAȚIONALĂ DE INFORMATICĂ
ETAPA JUDEȚEANĂ
CLASA A –X –A

Solutie_perle_c_10

```
#include <stdio.h>
#define NMAX 10000
int st[2][NMAX+10]; /* 1, 2, 3 e evident si 4, 5, 6 e A, B, C */
int main()
{
int t,l,v,i;
int niv[2];
int valid[2];
freopen("perle.in","r",stdin);
freopen("perle.out","w",stdout);
scanf("%d",&t);
while (t--)
      {
       scanf("%d",&l);
       if (l == 1)
          {scanf("%d",&v);
           printf("1\n");
           continue;         }
           else
           {
            valid[1] = valid[0] = 1;
            niv[1] = niv[0] = 1;
            st[0][0] = 5;
            st[1][0] = 6;
            while (l--)
                  {
                   scanf("%d",&v);
                    for (i = 0; i < 2; i++) /* trist da doar 2 valori
vreau */
                        {
                         if (valid[i])
                            {
                             if (st[i][niv[i]-1] < 4)
                               {
                                valid[i] = (st[i][--niv[i]]==v);
                                continue;
                               }

                             if (st[i][niv[i]-1] == 4)
                                 {
                                  niv[i]--;
                                  continue;
                                 }

                            if (st[i][niv[i]-1]==5)
                               {
```

```
                                if (v == 3) valid[i] = 0;
                                if (v == 1)
                                   {
                                    niv[i]--;
                                    st[i][niv[i]++] = 6;
                                    st[i][niv[i]++] = 4;
                                    st[i][niv[i]++] = 3;
                                    st[i][niv[i]++] = 4;
                                   }
                               /* 2-ul e ok :D */
                               continue;
                              }

                         /* e C in stiva */
                        if (v == 2) niv[i]--;
                        if (v == 1)
                           {
                             niv[i]--;
                             st[i][niv[i]++] = 4;
                             st[i][niv[i]++] = 2;
                            }
                      if (v == 3)
                         {
                           niv[i]--;
                           st[i][niv[i]++] = 6;
                           st[i][niv[i]++] = 5;
                          }

                   }

              if (!niv[i] && l) valid[i] = 0;
                                }
                         }

       printf("%d\n",((!niv[0] && valid[0]) || (!niv[1] && valid[1])));
           }
}


        return 0;
}
```

Solutie_rj_10

```
#include <fstream.h>
#define InFile "rj.in"
#define OutFile "rj.out"
#define NMax 102
#define NV 8

int n, m, xr, yr, xj, yj;
int dl[NV]={0, 1, 0, -1, -1, 1, -1, 1};
int dc[NV]={1, 0, -1, 0, -1, 1,  1,-1};
char l[NMax][NMax];
int r[NMax][NMax];

void citire(void);
void afisare(int [NMax][NMax]);
void parcurge (int, int, int[NMax][NMax]);

void main()
{int j[NMax][NMax];
 citire();
 parcurge(xr, yr, r);
 parcurge(xj, yj, j);
 afisare(j);
}

void citire(void)
{
int i, k;
char c;
ifstream f(InFile);
f>>n>>m;
for (i=0; i<=n+1; i++) l[i][0]=l[i][m+1]='X';
for (i=0; i<=m+1; i++) l[0][i]=l[n+1][i]='X';
f.get(c);
for (i=1; i<=n; i++)
        {for (k=1; k<=m; k++)
                {f.get(c); l[i][k]=c;
                 if (l[i][k]=='R') {xr=i; yr=k; l[i][k]=' ';}
                 if (l[i][k]=='J') {xj=i; yj=k; l[i][k]=' ';}
                }
        f.get(c);}
f.close();
}

void parcurge (int x0, int y0, int d[NMax][NMax])
{
struct Punct {int l, c;} C[NMax*NMax], p;
int inc=0, sf=0, i, k;
for (i=0; i<=n+1; i++)
        for (k=0; k<=m+1; k++) d[i][k]=-1;
C[0].l=x0; C[0].c=y0; d[x0][y0]=1;
while (inc<=sf)
        {
         p=C[inc++];
```

```
         for (i=0; i<NV; i++)
                if (l[p.l+dl[i]][p.c+dc[i]]==' ' &&
d[p.l+dl[i]][p.c+dc[i]]==-1)
                        {
                        d[p.l+dl[i]][p.c+dc[i]]=1+d[p.l][p.c];
                        C[++sf].l=p.l+dl[i];
                        C[sf].c=p.c+dc[i];
                        }
        }
}

void afisare(int j[NMax][NMax])
{
ofstream f(OutFile);
int tmin=NMax*NMax+5, xmin=-1, ymin=-1, i, k;
for (i=1; i<=n; i++)
        for (k=1; k<=m; k++)
                if (r[i][k]==j[i][k])
                        if (r[i][k]<tmin && r[i][k]!=-1)
                                {tmin=r[i][k]; xmin=i; ymin=k;}
f<<tmin<<' '<<xmin<<' '<<ymin<<endl;
f.close();
}
```