24ος ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΝΔΕΙΚΤΙΚΕΣ ΑΠΑΝΤΗΣΕΙΣ Γ΄ ΦΑΣΗΣ

Θέμα 1ο: Λουτράκι [30 Μονάδες]

C++ **Παναγιώτου Σωτήριος 59ο ΓΕΛ Αθηνών**

```
/*
USER:pdp24u226
TASK:loutraki
LANG:C++
*/

#include<bitset>
#include<set>
#include<map>
#include<queue>
#include<utility>
#include<algorithm>
#include<stack>

using namespace std;

#include"stdio.h"
#include"string.h"
#include"stdlib.h"
#include"assert.h"

FILE *fin,*fout;

int minx[200001],miny[200001];
int minx_bel[200001],miny_bel[200001];

bool bouno[1000001],paralia[1000001];

int N;

int main(){

      int i,j,k;

      fin=fopen("loutraki.in","r");
      assert(fin);

      fscanf(fin,"%d",&N);
      for(i=1;i<=N;i++){
            fscanf(fin,"%d %d",&j,&k);
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☏ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
            //printf("check (%d,%d)->(%d,
%d)\n",j,k,j+100000,k+100000);

            j+=100000;
            k+=100000;

            //min x for row

            if(minx_bel[k]==0||minx[k]>j){
                  minx_bel[k]=i;
                  minx[k]=j;
            }

            //min y for row

            if(miny_bel[j]==0||miny[j]>k){
                  miny_bel[j]=i;
                  miny[j]=k;
            }

      }

      fclose(fin);

      for(i=0;i<=200000;i++){
            if(minx_bel[i]!=0){
                  //printf("minx for row %d is %d, belongs to
%d\n",i-100000,minx[i]-100000,minx_bel[i]);
                  bouno[minx_bel[i]]=true;
            }
      }

      for(i=0;i<=200000;i++){
            if(miny_bel[i]!=0){
                  //printf("miny for col %d is %d, belongs to
%d\n",i-100000,miny[i]-100000,miny_bel[i]);
                  paralia[miny_bel[i]]=true;
            }
      }


      j=0;
      for(i=1;i<=N;i++){
            if(bouno[i]&&paralia[i]){
                  //printf("%d wins\n",i);
                  j++;
            }
      }

      fout=fopen("loutraki.out","w");
      fprintf(fout,"%d\n",j);
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
        fclose(fout);
        return 0;

}
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☏ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

Θέμα 2ο: Παλίνδρομο [30 Μονάδες]

C++ Ευάγγελος Κηπουρίδης 11ο ΓΕΛ Θες/νίκης

```
/*
USED : pdp24u11
TASK : minpali
LANG : C++
*/
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <string>
#include <algorithm>
#include <utility>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <stack>
#include <queue>

using namespace std;

#define MAXC 10000005
#define mp make_pair
#define INF 0x3f3f3f3f
#define X first
#define Y second

typedef long long llint;
typedef pair<int,int> pii;

int N, F[MAXC];
char str[MAXC], rev[MAXC];

void Read() {
      int i;

      scanf ("%d %s\n", &N, str);
      for ( i=0; i<N; ++i ) {
            rev[i] = str[N-i-1];
      }
}

void Prefix () {
      int i, j;

      F[0] = F[1] = 0;
      for ( i=2; i<=N; ++i ) {
            j = F[i-1];
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☏ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
            while ( 1 ) {
                  if ( rev[i-1] == rev[j] ) {
                        F[i] = j+1;
                        break;
                  }
                  else if ( j==0 ) {
                        F[i] = j;
                        break;
                  }
                  else {
                        j = F[j];
                  }
            }
      }
}
void KMP () {
      int i=0, j=0;

      while ( 1 ) {
            if ( i==N ) {
                  break;
            }

            if ( str[i] == rev[j] ) {
                  ++i;
                  ++j;
            }
            else if ( j==0 ) {
                  ++i;
            }
            else {
                  j = F[j];
            }
      }

      printf ("%d\n", 2*N - j );
}

int main ( void ) {
#ifdef D
      freopen ("input","r",stdin);
#endif
      freopen ("minpali.in","r",stdin);
      freopen ("minpali.out", "w", stdout );

      Read();
      Prefix();
      KMP();

      return 0;
}
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

Θέμα 3ο: Σουβλάκια [40 Μονάδες]

C++ Ευάγγελος Κηπουρίδης 11ο ΓΕΛ Θες/νίκης (90%)

```
/*
USED : pdp24u11
TASK : souvlakia
LANG : C++
*/
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <string>
#include <algorithm>
#include <utility>
#include <vector>
#include <list>
#include <set>
#include <map>
#include <queue>

using namespace std;

#define MAXN 100005
#define mp make_pair
#define SAFE 60000000LL
#define INF 0x3f3f3f3f
#define X first
#define Y second

typedef long long llint;
typedef pair<int,int> pii;

struct Node {
      int d[3];
      int id;
} node[MAXN];

int N, M, A, B, C, L, in[MAXN], nstack, stack[MAXN];
vector<int> e[MAXN], dist[MAXN];
set< pii > coast;
set< pii >::iterator pos, pos2;

bool comp ( Node a, Node b ) {
      if ( a.d[0] < b.d[0] ) {
            return true;
      }
      if ( a.d[0] == b.d[0] && a.d[1] < b.d[1] ) {
            return true;
      }
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☏ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
      if ( a.d[0] == b.d[0] && a.d[1] == b.d[1] && a.d[2] <
b.d[2] ) {
            return true;
      }
      return false;
}

void Read() {
      int i, a, b, d;

      scanf ("%d %d", &N, &M);

      for ( i=1; i<=M; ++i ) {
            scanf ("%d %d %d", &a, &b, &d);
            e[a].push_back(b);
            e[b].push_back(a);
            dist[a].push_back(d);
            dist[b].push_back(d);
      }
      scanf ("%d %d %d %d", &A, &B, &C, &L);
      for ( i=1; i<=N; ++i ) {
            node[i].d[0] = node[i].d[1] = node[i].d[2] = INF;
            node[i].id = i;
      }
}

void Print () {
      int i, a;

      for ( i=1; i<=L; ++i ) {
            scanf ("%d", &a);
            if ( in[a] ) {
                  printf ("YES\n");
            }
            else {
                  printf ("NO\n");
            }
      }
}

void Dijkstra ( int s, int d ) {
      int v, i;
      node[s].d[d] = 0;
      set< pii > S;
      S.insert ( mp(0,s) );


      while ( !S.empty() ) {
            v = S.begin()->Y;
            S.erase ( S.begin() );
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
            for ( i=0; i<e[v].size(); ++i ) {
                  if ( node[ e[v][i] ].d[d] > node[v].d[d] +
dist[v][i] ) {
                        if ( node[ e[v][i] ].d[d] != INF ) {
                              S.erase ( S.find ( mp ( node[ e[v]
[i] ].d[d], e[v][i] ) ) );
                        }
                        node[ e[v][i] ].d[d] = node[v].d[d] +
dist[v][i];
                        S.insert ( mp ( node[ e[v][i] ].d[d],
e[v][i] ) );
                  }
            }

      }
}

bool Ok ( set< pii >::iterator thesi, int i ) {
      --thesi;
      // thesi->X < node[i].d[1]
      if ( thesi->Y < node[i].d[2] ) {
            return false;
      }
      return true;
}

void Insert ( int i ) {
      int new_val;

      pos = coast.lower_bound ( mp ( node[i].d[1], -1 ) );

      if ( pos == coast.begin() || Ok(pos, i) ) {
            if ( pos != coast.end() && pos->X == node[i].d[1] )
{
                  new_val = min ( pos->Y, node[i].d[2] );
                  coast.erase ( coast.find ( mp ( pos->X, pos->Y
) ) );
            }
            else {
                  new_val = node[i].d[2];
            }
            coast.insert ( mp ( node[i].d[1], new_val ) );
            pos = pos2 = coast.find ( mp ( node[i].d[1], new_val
) );
            for ( ++pos2, ++pos; pos2 != coast.end(); ++pos2 ) {
                  if ( ! ( new_val <= pos2->Y ) ) {
                        break;
                  }
            }

            if ( pos != coast.end() && pos != pos2 ) {
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
                coast.erase ( pos, pos2 );
            }
        }
}

void Compute() {
        int i, j;

        sort ( node+1, node+N+1, comp );
        in[ node[1].id ] = 1;
        coast.insert ( mp ( node[1].d[1], node[1].d[2] ) );

        nstack = 0;
        for ( i=2; i<=N; ++i ) {
            if ( node[i].d[0] != node[i-1].d[0] ) {
                break;
            }
            in[ node[i].id ] = 1;
            stack[ ++nstack ] = i;
        }

        for (  ; i<=N; ++i ) {
            if ( node[i].d[0] != node[i-1].d[0] ) {
                for ( j=1; j<=nstack; ++j ) {
                    Insert ( stack[j] );
                }
                nstack = 0;
            }

            pos = coast.lower_bound ( mp ( node[i].d[1], -1 ) );

#ifdef D
            for ( pos2 = coast.begin(); pos2 != coast.end(); +
+pos2 ) {
                printf ("%d %d\n", pos2->X, pos2->Y);
            }
            printf ("\n");
#endif

            if ( pos == coast.begin() || Ok(pos, i) ) {
                in[ node[i].id ] = 1;
                stack[ ++nstack ] = i;
            }
        }

}

void Brute_Compute() {
        int i, j, no, a;

        for ( i=1; i<=L; ++i ) {
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr

```
            scanf ("%d", &a);
            no = 0;
            for ( j=1; j<=N; ++j ) {
                  if ( node[j].d[0] < node[a].d[0] &&
node[j].d[1] < node[a].d[1] && node[j].d[2] < node[a].d[2] ) {
                        printf ("NO\n");
                        no = 1;
                        break;
                  }
            }
            if ( no==0 ) {
                  printf ("YES\n");
            }
      }

}

int main ( void ) {
#ifdef D
      freopen ("input","r",stdin);
#endif
      freopen ("souvlakia.in","r",stdin);
      freopen ("souvlakia.out","w",stdout);

      Read();
      Dijkstra ( A, 0 );
      Dijkstra ( B, 1 );
      Dijkstra ( C, 2 );
      if ( (llint)N*(llint)L <= SAFE ) {
            Brute_Compute();
      }
      else {
            Compute();
            Print();
      }

      return 0;
}
```


Copyright ΕΠΥ 2011-12. Στουρνάρη 37, Αθήνα 106 82 Stournari Str 37 PC 106 82, Athens, Greece
☎ +30-210-3300999, 🖷+30-210-3301893 E-mail: epy@epy.gr, Web: www.epy.gr