



25^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Ενδεικτικές Λύσεις

Triangle

C++

Χρίστογλου Γεώργιος

14ο ΓΕΛ Θεσσαλονίκης

```
/*
User:u2539
LANG:C++
TASK:triangle
*/

#include<stdio.h>
#include<string.h>
#include<vector>
#include<set>
#include<queue>
#include<algorithm>
#include<string>
#include<cmath>

using namespace std;

#define MAXN 1005
#define f first
#define s second
#define mp make_pair
#define INF 0x3f3f3f3f
#define pii pair<int,int>
#define BASE 256
#define MOD 7293847562347896LL

int N,dp[MAXN][MAXN],back[MAXN][MAXN];

void Read () {
    scanf("%d",&N);

    for (int i=1;i<=N;i++) {
        for (int j=1;j<=i;j++) {
            scanf("%d",&back[i][j]);
        }
    }
}
```

Σελίδα 1 από 5



```
    }
}

void Solve() {

    dp[1][1]=back[1][1]; // base cases of dp

    for (int i=2;i<=N;i++) {
        for (int j=1;j<=i;j++) {
            if (j!=i)
                dp[i][j]=max(dp[i][j], dp[i-1][j]+back[i][j]);
            if (j!=1)
                dp[i][j]=max(dp[i][j], dp[i-1][j-1]+back[i]
[j]);
        }
    }

    int ans=0;

    for (int i=1;i<=N;i++)
        ans=max(dp[N][i], ans);

    printf("%d\n", ans);
}

int main () {
    freopen("triangle.in", "r", stdin);
    freopen("triangle.out", "w", stdout);

    Read();
    Solve();

    return 0;
}
```

Sound

C++

Σελίδα 2 από 5



Λώς Δημήτριος

Κολλέγιο Ψυχικού

```
#include <cstdio>
#include <cstdlib>
#include <list>
#define MAXN 1000100
#define mk(a, b) make_pair(a, b)
#define F first
#define S second
using namespace std;
int A[MAXN];
int main(){
    list< pair<int, int> > minim, maxim;
    int n, m, c;
    FILE *fi = fopen("sound.in", "r");
    fscanf(fi, "%d%d%d", &n, &m, &c);
    for (int i = 0; i!=n; i++) fscanf(fi, "%d", &A[i]);
    m--;
    for (int i = 0; i<m; i++){
        while(!minim.empty() && A[i]<=minim.front().F) minim.-
pop_front();
        minim.push_front(mk(A[i], i));
        while(!maxim.empty() && A[i]>=maxim.front().F) maxim.-
pop_front();
        maxim.push_front(mk(A[i], i));
    }
    m++;
    bool found = false;
    FILE *fo = fopen("sound.out", "w");
    for (int i = m-1; i<n; i++){
        // Bgazoume auta pou exei perasei h epirroh tous
        if (minim.back().S < i-m+1 && !minim.empty())
minim.pop_back();
        if (maxim.back().S < i-m+1 && !maxim.empty())
maxim.pop_back();
        // Update monotony
        while(!minim.empty() && A[i]<=minim.front().F) minim.-
pop_front();
        minim.push_front(mk(A[i], i));
        while(!maxim.empty() && A[i]>=maxim.front().F) maxim.-
pop_front();
        maxim.push_front(mk(A[i], i));
        if (maxim.back().F - minim.back().F<=c) {
            found = true;
            fprintf(fo, "%d\n", i - m+2); // +2 gt einai bash 1
        }
    }
    if (!found) fprintf(fo, "NULL\n");
    return 0;
}
```

Σελίδα 3 από 5



}

Telecom

C++

Παναγιώτης Κωστοπαναγιώτης

2ο ΓΕΛ Πεύκης

```
/*
USER: u252
LANG: C++
TASK: TELECOM
*/
#include <cstdio>
#include <algorithm>

#define MAX 500000

using namespace std;

long long p[MAX+1], num[MAX+1], rank[MAX+1];

struct edge {
    int u, v, w;
};

void makenode(int u)
{
    p[ u ] = u;
    num[ u ] = 1;
    rank[ u ] = 1;
}

int find(int u)
{
    if(u == p[ u ]) {
        return u;
    } else {
        return p[ u ] = find(p[ u ]);
    }
}

void Union(int u, int v)
{
    u = find( u );
    v = find( v );
    if( rank[ u ] > rank[ v ] ) {
        p[ v ] = u;
        num[ u ] += num[ v ];
    }
}
```

Σελίδα 4 από 5



```
    } else {
        p[ u ] = v;
        num[ v ] += num[ u ];
        if( rank[ u ] == rank[ v ] ) {
            rank[ v ]++;
        }
    }
}

bool cmp( edge a, edge b )
{
    return a.w < b.w;
}

int main( void )
{
    freopen("telecom.in","rt",stdin);
    freopen("telecom.out","wt",stdout);
    int N;
    unsigned long long cost = 0;
    scanf("%d",&N);
    edge E[N-1];
    for(int i = 0; i < N - 1; i++ ) {
        scanf("%d%d%d",&E[ i ].u,&E[ i ].v,&E[ i ].w);
        makenode( E[ i ].u );
        makenode( E[ i ].v );
    }
    sort(E, E+N-1, cmp);
    for(int i = 0; i < N-1; i++ ) {
        int fu = find( E[ i ].u ),fv = find( E[ i ].v );
        if( fu != fv ) {
            cost += num[ fu ]*num[ fv ]*(E[ i ].w + 1);
            Union( E[ i ].u , E[ i ].v );
        }
    }
    printf("%llu\n",cost-N+1);
    return 0;
}
```