

# BILTEN

# 2015.

## 3. KANTONALNO TAKMIČENJE UČENIKA SREDNJIH ŠKOLA TUZLANSKOG KANTONA IZ INFORMATIKE

JU Mješovita srednja škola "Hasan Kikić" Gradačac,  
14.03.2015.



# JU MSŠ „Hasan Kikić“

## 3. KANTONALNO TAKMIČENJE UČENIKA SREDNJIH ŠKOLA TUZLANSKOG KANTONA IZ INFORMATIKE

Javna ustanova MSŠ „Hasan Kikić“ Gradačac je savremena odgojno-obrazovna ustanova za stručno i tehničko obrazovanje mladih sa tradicijom dugom preko 50 godina.

Danas profesionalno osposobljavamo učenike različitih zvanja i zanimanja u tehničkim i stručnim školama:

- mašinska,
- tekstilna,
- elektrotehnička,
- poljoprivredna,
- medicinska,
- građevinska,
- ekonomska,
- ugostiteljsko-turistička,
- saobraćajna, drvoprerađivačka,
- škola uslužnih djelatnosti.

Sastavni dio odgojno-obrazovnog procesa učenika je i angažman u raznovrsnim sekcijama i drugim oblicima vannastavnih aktivnosti.

Škola je u reformi  
stručnog i  
tehničkog  
obrazovanja  
prihvatila  
PHARE-VET,  
EU VET I GTZ  
program po  
Standardima  
Evropske unije.

*Škola je dobila ime po bosanskohercegovački književniku i revolucionaru Hasanu Kikiću.*

## JU MSŠ „Hasan Kikić“ Gradačac

Izgradnjom nove zgrade JU MSŠ “Hasan Kikić” dobiva adekvatan prostor sa fiskulturnom salom. Nastava se u novoj zgradi odvija od oktobra 1999. godine. Danas je ova škola opremljena savremenim učionicama i kabinetima za informatiku. U ovom moderno uređenom objektu učenicima je pružena mogućnost da stiču znanja uz savremena nastavna sredstva i pomagala.



## Općina Gradačac



Područje Gradačca se u historijskim izvorima prvi put javlja 1326-29. kao srednjovjekovna župa Nenavište. Kasnije se na ovom području javlja grad Gračac (1461). Pod Osmanlijskom je vlašću od 1519. Gračačka nahija spomenuta u jednom defteru iz 1533. god. 1634. navodi se da Gradačac pripada Gračaničkom kadiluku. To je ujedno i prvi dokumenat u kojem se ovo mjesto javlja pod sadašnjim imenom. Nakon Karlovačkog mira (1699.) Gradačac, kojem je tada bila namijenjena odbrana osmanlijskih granica na Savi, mnogo dobija na svom strateškom značaju. Godine 1701. postaje palanka, a od 1710. sjedište kapetanije. Poslije toga varošica se brzo razvijala, pa je polovinom 19. vijeka imala dva hana, dvije medrese, četiri džamije, nekoliko mekteba i oko 40 trgovačkih i zanatskih radnji. Najveća zasluga za takav prosperitet mjesta pripada gradačačkim nasljednim kapetanima iz obitelji Gradašćevića: Osman-kapetanu i Murat-kapetanu, a naročito predvodniku vojske bosanskih feudalaca u borbi protiv osmanlijske uprave 1830. godine, Husein-kapetanu Gradašćeviću, poznatom pod imenom "Zmaj od Bosne".

**Spisak učesnika III Kantonalnog takmičenja iz informatike**

<i>R.b.</i>	<i>Prezime i ime takmičara</i>	<i>Škola iz koje dolazi</i>	<i>Voditelj</i>
1	JUKAN AJNUR	JU Gimnazija " Mustafa Novalić" Gradačac	Kostić Hasanović Brankica
2	BUNGUR AIDA		Suljkanović Jasmin
3	STEJEPIĆ ĆOSIĆ VALENTIN	JU Mješovita srednja elektrotehnička škola Tuzla	Imširović Edin
4	HASANBAŠIĆ AMIR		Bojić Dejan
5	ISABEGOVIĆ MUHAMED		Zahirović Melisa
6	IMAMOVIĆ MIRZA		
7	HASANBAŠIĆ SEDIN		
8	KARIĆ ALDIN	JU Gimnazija Lukavac	Mujić Bernisa
9	MUJKIĆ AHMED		
10	ĆATIĆ BENJAMIN		
11	SRABOVIĆ NEDIM	Gimnazija "Dr.Mustafa Kamarić" Gračanica	Husičić Nedžad
12	HADŽIĆ HARIS		
13	SOLJANKIĆ JASMINA	Građevinska škola Tuzla	Kostić Alma
14	GAZDIĆ RIFET	JU MSŠ "Musa Ćazim Ćatić" Kladanj	Gazdić Senahid
15	RAŠIDOVIĆ KENAN		
16	HALILOVIĆ SUAD	JU MS Elektro-mašinska škola Lukavac	Dželilović Marizela
17	GRADAŠKIĆ DINO		
18	NUKIĆ ALMIR	JU MSŠ Saobraćajna škola Tuzla	Junuzagić Adisa
19	SLANJANKIĆ TARIK	JU MSŠ Srebrenik	Ordagić Mirzet
20	JUKIĆ DŽEMAL		Nišić Damir
21	BEGIĆ HARIS	JU MS Hemijska škola Tuzla	Bećirović Enes
22	VUGDALIĆ AHMED		
23	ČAUŠEVIĆ SAJIB	JU Mješovita srednja škola Gračanica	Pejić Sladjana
24	SPREČIĆ ANES		
25	MALEŠKIĆ NASIHA	JU Mješovita srednja škola Doboj Istok	Bećirović Alma
26	BAČINOVIĆ SAID		

27	PODGORČIĆ ENVER	Gimnazija Živinice	Lučić Ilija
28	IBRAHIMOVIĆ AMRA		
29	OMEROVIĆ MUHAMED		
30	ABDULAHOVIĆ ADNAN	Mješovita srednja škola Teočak	
31	ABDULAHOVIĆ BELMIN ( rez. )		
32	KOVAČEVIĆ NERMIN	JU Mješovita srednja škola Banovići	Bećirović Alma
33	KOZAREVIĆ NADIL	JU Behram - Begova Medresa Tuzla	Kozarević Elvedin
34	KRDŽALIĆ AMILA		
35	BAJIĆ ADNAN	Gimnazija "Meša Selimović" Tuzla	Fočić Muhidin
36	MUJAČIĆ AMIR		Harčin Fuad
37	JAGODIĆ EMIR		Šehović Mirela
38	IMAMOVIĆ FARUK		
39	HASIĆ OMAR		
40	ŠIŠIĆ HARIS	Gimnazija "Ismet Mujezinović" Tuzla	Imamović Hajrudin
41	MUKIĆ EDIN		
42	ŠEHOVIĆ DARIO	Opća gimnazija KŠC "Sv. Franjo" Tuzla	Andrejević Andrej
43	ĐONLAGIĆ AZUR		
44	JURKOVIĆ GABRIJEL		
45	KOVAČEVIĆ IGOR		
46	CVITKUŠIĆ MIŠEL	MSŠ "Hasan Kikić" Gradačac	Sarajlić Nazifa
47	HRŠTIĆ MUHAMED		
48	HASIĆ KENAN		
49	BEGOVIĆ FATMIR		

14.03.2015. u MSS „Hasan Kikić“ u Gradačcu na III Kantonalnom takmičenju iz informatike učestvovalo je 49 učenika iz 20 srednje škole Tuzlanskog kantona.



## Zadatak - TORNJIVI

Na mapi  $R \times S$  se odvija napad. Napadači su bosonogi pljačkaši, a od njih se branimo malim topovima koji se nalaze na malim drvenim tornjevima.

Na svakom tornju nalaze se po dva topa, spojena tako da uvijek pucaju u medjusobno okomitim smjerovima.

Preciznije, topovi na jednom tornju mogu biti u jednoj od sljedeće četiri konfiguracije:

1. Pucaju lijevo i dolje;
2. Pucaju dolje i desno;
3. Pucaju desno i gore;
4. Pucaju gore i lijevo.

Topovska kugla koja pogodi napadača raznese ga te nastavi letjeti u istom smjeru. Kugla koja pogodi dvorac zaustavlja se i ne nanosi mu nikakvu štetu (jer su dvorci veliki i jaki).

Medjutim, kad bi kugla pogodila toranj, raznijela bi ga (jer su tornjevi mali i krhki).

Potrebno je okrenuti topove na tornjevima tako da, kada ispalimo točno po jedan hitac iz svakog topa, raznesemo sve napadače, a svi tornjevi ostanu čitavi.

### Ulazni podaci

U prvom redu nalaze se dva prirodna broja  $R$  i  $S$  ( $1 \leq R, S \leq 100$ ), dimenzije mape.

U sljedećih  $R$  redova nalazi se po  $S$  znakova koji opisuju mapu.

Svaki od znakova može biti veliko slovo 'T' (toranj), malo slovo 'n' (napadač), znak '#' (dvorac) ili znak '.' (prazno).

Napomena:

Ulazni podaci će biti takvi da će rješenje, iako ne nužno jedinstveno, uvijek postojati.

### Izlazni podaci

U  $R$  redova ispišite mapu u istom formatu kao u ulazu koja dodatno sadrži orijentaciju topova na svakom od tornjeva. Točnije, na pozicijama tornjeva potrebno je ispisati jednu od znamenki '1', '2', '3' ili '4', konfiguraciju topova na tom tornju kako je opisano u tekstu zadatka



**Ulaz**  
 5 9  
 .n..T..n.  
 .T..n....  
 .n.#..n.  
 ....n..T.  
 .n..T..n.

**Izlaz**  
 .n..4..n.  
 .2..n....  
 .n.#..n.  
 ....n..4.  
 .n..3..n.

**Ulaz**  
 8 8  
 nn.nnTnn  
 nnnnnnT.  
 nn.nTnnn  
 nn.Tnnnn  
 Tn..##..  
 nnnn.nnT  
 nnT..nnn  
 .Tn#.Tnn

**Izlaz**  
 nn.nn3nn  
 nnnnnn1.  
 nn.n3nnn  
 nn.3nnnn  
 4n..##..  
 nnnn.nn1  
 nn4..nnn  
 .3n#.4nn

**Ulaz**  
 20 20  
 TT#n#nnnn....nn.Tnnn  
 TT#n#nnnT.#.....  
 ##TT#nTnn.#nnTn.n...  
 nnTT#Tnnn.nnnnn.nnnn  
 ####T..Tn.nnnnn.nnnn  
 nnn..nnnn.nnT.n##...  
 Tnn..nnnn.nnnnnnnnnn  
 nTn#.nnnn.nTn.nnn...  
 ##nn.nnnn.nnnnnTn...  
 Tn.n##...Tn##Tnnnnn  
 .n.n.....nnTnnnnnnn  
 .n.nTnnn##nn....Tnnn  
 .n.nn...Tnnnnnnnnnnn  
 nnnnnT#.n.nn.....Tn  
 .n.nnn..n.nn.Tnnnnnn  
 nT.nnn#nnnnnnnnnnTn.  
 nnnTnn..n.nn.n...nn.  
 ..Tnnn.nnnnnnnnnnnn  
 ..n.nn..n.nn.n...nnT  
 nnnnnn.nnTnn.n...nnn

**Izlaz**  
 43#n#nnnn....nn.2nnn  
 12#n#nnn4.#.....  
 ##43#n3nn.#nn4n.n...  
 nn12#3nnn.nnnnn.nnnn  
 ####4..3n.nnnnn.nnnn  
 nnn..nnnn.nn1.n##...  
 3nn..nnnn.nnnnnnnnnn  
 n4n#.nnnn.n1n.nnn...  
 ##nn.nnnn.nnnnn4n...  
 4n.n##...1n##3nnnnn  
 .n.n.....nn3nnnnnnn  
 .n.n2nnn##nn....3nnn  
 .n.nn...2nnnnnnnnnnn  
 nnnnn1#.n.nn.....2n  
 .n.nnn..n.nn.2nnnnnn  
 n4.nnn#nnnnnnnnnn1n.  
 nnn4nn..n.nn.n...nn.  
 ..3nnn.nnnnnnnnnnnn  
 ..n.nn..n.nn.n...nn1  
 nnnnnn.nn4nn.n...nnn

/\*

```

Zadatak TORNJIVI
*/
#include <algorithm>
#include <cstdio>
#include <vector>
#include <cstring>
#include <iostream>
using namespace std;
typedef pair<int,int> par;
#define MAX 100
int n;
int rr[MAX*MAX], ss[MAX*MAX];
vector<par> adj[2*MAX*MAX][2];
int R, S;
char ulaz[MAX][MAX+1];
int broj[MAX][MAX];
int nadji[MAX][MAX][4];
int cookie[2*MAX*MAX], COOKIE;
char valid_cookie[4*MAX*MAX+2];
int var[2*MAX*MAX];
int dfs( int X, int x ) {
    if( valid_cookie[cookie[X]] ) return var[X] == x;
    cookie[X] = COOKIE;
    var[X] = x;
    x ^= 1;
    for( vector<par>::iterator it = adj[X][x].begin(); it != adj[X][x].end(); ++it )

```

```

    if( !dfs( it->first, it->second ) )
        return 0;
return 1;
}
int main( void ) {
    freopen ( "tornjevi.in", "r", stdin );//aa
    freopen ( "tornjevi.out", "w", stdout );//aa
    scanf( "%d%d", &R, &S );
    for( int r = 0; r < R; ++r ) {
        scanf( "%s", ulaz[r] );
        for( int s = 0; s < S; ++s ) {
            if( ulaz[r][s] == 'T' ) {
                rr[n] = r;
                ss[n] = s;
                broj[r][s] = n++;
            }
        }
    }
    memset( nadji, -1, sizeof nadji );
    for( int r = 0; r < R; ++r )
        for( int s = 0; s < S; ++s ) {
            if( ulaz[r][s] == '#' ) continue;
            if( r-1 >= 0 ) nadji[r][s][0] = ulaz[r-1][s] == 'T' ? broj[r-1][s] : nadji[r-1][s][0];
            if( s-1 >= 0 ) nadji[r][s][1] = ulaz[r][s-1] == 'T' ? broj[r][s-1] : nadji[r][s-1][1];
        }
    for( int r = R-1; r >= 0; --r )

```

```

for( int s = S-1; s >= 0; --s ) {
    if( ulaz[r][s] == '#' ) continue;
    if( r+1 < R ) nadji[r][s][2] = ulaz[r+1][s] == 'T' ? broj[r+1][s] : nadji[r+1][s][2];
    if( s+1 < S ) nadji[r][s][3] = ulaz[r][s+1] == 'T' ? broj[r][s+1] : nadji[r][s+1][3];
}
try {
    memset( cookie, 0, sizeof cookie );
    memset( valid_cookie, 1, sizeof valid_cookie );
    valid_cookie[0] = 0;
    COOKIE = 1;
    for( int r = 0; r < R; ++r ) {
        for( int s = 0; s < S; ++s ) {
            if( ulaz[r][s] == 'T' ) {
                int x = broj[r][s];
                if( nadji[r][s][0] != -1 && nadji[r][s][2] != -1 ) throw 1;
                if( nadji[r][s][1] != -1 && nadji[r][s][3] != -1 ) throw 1;
                if( nadji[r][s][0] != -1 ) { var[2*x+0] = 1; cookie[2*x+0] = COOKIE; }
                if( nadji[r][s][1] != -1 ) { var[2*x+1] = 1; cookie[2*x+1] = COOKIE; }
                if( nadji[r][s][2] != -1 ) { var[2*x+0] = 0; cookie[2*x+0] = COOKIE; }
                if( nadji[r][s][3] != -1 ) { var[2*x+1] = 0; cookie[2*x+1] = COOKIE; }
            }
            if( ulaz[r][s] == 'x' || ulaz[r][s] == 'n' ) {
                int V = nadji[r][s][0], v = 1; if( V == -1 ) { V = nadji[r][s][2]; v = 0; }
                int H = nadji[r][s][1], h = 1; if( H == -1 ) { H = nadji[r][s][3]; h = 0; }
                if( V == -1 && H == -1 ) throw 1;
                if( V == -1 && H != -1 ) { var[2*H+1] = h; cookie[2*H+1] = COOKIE; }
            }
        }
    }
}

```

```

    if( V != -1 && H == -1 ) { var[2*V+0] = v; cookie[2*V+0] = COOKIE; }

    if( V != -1 && H != -1 ) {
        adj[2*H+1][h].push_back( par( 2*V+0, v ) );
        adj[2*V+0][v].push_back( par( 2*H+1, h ) );
    }
}
}
}

for( int i = 0; i < 2*n; ++i ) {
    if( valid_cookie[cookie[i]] ) continue;
    ++COOKIE;
    if( dfs( i, 0 ) ) continue;
    valid_cookie[COOKIE] = 0;
    ++COOKIE;
    if( dfs( i, 1 ) ) continue;
    valid_cookie[COOKIE] = 0;
    throw 1;
}

for( int i = 0; i < n; ++i ) {
    if( var[2*i] == 0 && var[2*i+1] == 0 ) ulaz[rr[i]][ss[i]] = '4';
    if( var[2*i] == 0 && var[2*i+1] == 1 ) ulaz[rr[i]][ss[i]] = '3';
    if( var[2*i] == 1 && var[2*i+1] == 0 ) ulaz[rr[i]][ss[i]] = '1';
    if( var[2*i] == 1 && var[2*i+1] == 1 ) ulaz[rr[i]][ss[i]] = '2';
}

for( int r = 0; r < R; ++r ) printf( "%s\n", ulaz[r] );
} catch( ... ) {

```

```

    printf( "NEMA\n" );
}
return 0;
}

(*
Zadatak TORNJEVI
*)

program tornjevi;
const maxn = 100;
var
    n, R, S    : longint;
    rr, ss    : array[1..maxn*maxn] of longint;
    ulaz      : array[1..maxn] of string;
    broj      : array[1..maxn, 1..maxn] of longint;
    najblizi  : array[1..maxn, 1..maxn, 0..3] of longint;
    cookie    : array[0..2*maxn*maxn] of longint;
    sadcookie : longint;
    valid_cookie : array[0..4*maxn*maxn+1] of boolean;
    vari      : array[0..2*maxn*maxn] of boolean;
    adjcnt    : array[0..2*maxn*maxn, false..true] of longint;
    adjvar    : array[0..2*maxn*maxn, false..true, 1..maxn] of longint;
    adjvri    : array[0..2*maxn*maxn, false..true, 1..maxn] of boolean;

function dfs(varijabla : longint; vrijednost : boolean):boolean;
var i : longint;

```

```
begin
  if valid_cookie[cookie[varijabla]] then begin
    dfs := vari[varijabla] = vrijednost;
    exit;
  end;
  cookie[varijabla] := sadcookie;
  vari[varijabla] := vrijednost;

  vrijednost := not vrijednost;
  for i:=1 to adjcnt[varijabla, vrijednost] do
    if not dfs(adjvar[varijabla, vrijednost, i], adjvri[varijabla, vrijednost, i]) then begin
      dfs := false;
      exit;
    end;
  end;

  dfs := true;
end;
var
  i, j, k, x : longint;
  V, H      : longint;
  vvar, hvar : boolean;
begin
  n := 0;
  readln(R, S);
  for i:=1 to R do begin
    readln(ulaz[i]);
    for j:=1 to S do begin
```

```

if ulaz[i, j] = 'T' then begin
    rr[n] := i;
    ss[n] := j;
    broj[i, j] := n;
    n := n+1;
end;
end;
end;
for i:=1 to R do
    for j:=1 to S do
        for k:=0 to 3 do
            najblizi[i, j, k] := -1;
        for i:=1 to R do begin
            for j:=1 to S do begin
                if ulaz[i, j] = '#' then continue;
                if i > 1 then begin
                    if ulaz[i-1, j] = 'T' then
                        najblizi[i, j, 0] := broj[i-1, j]
                    else
                        najblizi[i, j, 0] := najblizi[i-1, j, 0];
                end;
                if j > 1 then begin
                    if ulaz[i, j-1] = 'T' then
                        najblizi[i, j, 1] := broj[i, j-1]
                    else
                        najblizi[i, j, 1] := najblizi[i, j-1, 1];
                end;
            end;
        end;
    end;
end;

```



```
    end;
  end;
end;
for i:=R downto 1 do begin
  for j:=S downto 1 do begin
    if ulaz[i, j] = '#' then continue;
    if i < R then begin
      if ulaz[i+1, j] = 'T' then
        najblizi[i, j, 2] := broj[i+1, j]
      else
        najblizi[i, j, 2] := najblizi[i+1, j, 2];
      end;
    if j < S then begin
      if ulaz[i, j+1] = 'T' then
        najblizi[i, j, 3] := broj[i, j+1]
      else
        najblizi[i, j, 3] := najblizi[i, j+1, 3];
      end;
    end;
  end;
end;
for i:=0 to 2*n-1 do begin
  cookie[i] := 0;
  valid_cookie[i] := true;
end;
valid_cookie[0] := false;
sadcookie := 1;
```

```

for i:=1 to R do begin
  for j:=1 to S do begin
    if ulaz[i, j] = 'T' then begin
      x := broj[i, j];
      if ((najblizi[i, j, 0] <> -1) and (najblizi[i, j, 2] <> -1)) or
        ((najblizi[i, j, 1] <> -1) and (najblizi[i, j, 3] <> -1)) then begin
        writeln('NEMA');
        exit;
      end;
      if najblizi[i, j, 0] <> -1 then begin vari[2*x+0] := true; cookie[2*x+0] := sadcookie; end;
      if najblizi[i, j, 1] <> -1 then begin vari[2*x+1] := true; cookie[2*x+1] := sadcookie; end;
      if najblizi[i, j, 2] <> -1 then begin vari[2*x+0] := false; cookie[2*x+0] := sadcookie; end;
      if najblizi[i, j, 3] <> -1 then begin vari[2*x+1] := false; cookie[2*x+1] := sadcookie; end;
    end;
    if (ulaz[i, j] = 'x') or (ulaz[i, j] = 'n') then begin
      V := najblizi[i, j, 0]; vvar := true; if V = -1 then begin V := najblizi[i, j, 2]; vvar := false; end;
      H := najblizi[i, j, 1]; hvar := true; if H = -1 then begin H := najblizi[i, j, 3]; hvar := false; end;
      if (V = -1) and (H = -1) then begin writeln('NEMA'); exit; end;
      if (V = -1) and (H <> -1) then begin vari[2*H+1] := hvar; cookie[2*H+1] := sadcookie; end;
      if (V <> -1) and (H = -1) then begin vari[2*V+0] := vvar; cookie[2*V+0] := sadcookie; end;
      if (V <> -1) and (H <> -1) then begin
        adjcnt[2*H+1, hvar] := adjcnt[2*H+1, hvar] + 1;
        adjvar[2*H+1, hvar, adjcnt[2*H+1, hvar]] := 2*V+0;
        adjvri[2*H+1, hvar, adjcnt[2*H+1, hvar]] := vvar;
        adjcnt[2*V+0, vvar] := adjcnt[2*V+0, vvar] + 1;
        adjvar[2*V+0, vvar, adjcnt[2*V+0, vvar]] := 2*H+1;
      end;
    end;
  end;
end;

```

```

        adjvri[2*V+0, vvar, adjcnt[2*V+0, vvar]] := hvar;
    end;
end;
end;
end;
for i:=0 to 2*n-1 do begin
    if valid_cookie[cookie[i]] then continue;
    sadcookie := sadcookie+1;
    if dfs(i, false) then continue;
    valid_cookie[sadcookie] := false;
    sadcookie := sadcookie+1;
    if dfs(i, true) then continue;
    valid_cookie[sadcookie] := false;
    writeln('NEMA');
    exit;
end;
for i:=0 to n-1 do begin
    if not vari[2*i] and not vari[2*i+1] then ulaz[rr[i], ss[i]] := '4';
    if not vari[2*i] and vari[2*i+1] then ulaz[rr[i], ss[i]] := '3';
    if vari[2*i] and not vari[2*i+1] then ulaz[rr[i], ss[i]] := '1';
    if vari[2*i] and vari[2*i+1] then ulaz[rr[i], ss[i]] := '2';
end;
for i:=1 to R do
    writeln(ulaz[i]);
end.

```

## Zadatak - TRENING

Amar ima digitalni sat kojeg je nedavno sam napravio. Svaka od četiri brojke na satu prikazana je pomoću mreže crvenih lampica poredanih u 5 redaka i 3 stupca. Brojevi su prikazani kombinacijama upaljenih i ugašenih lampica kao na slici (znak '#' predstavlja upaljenu, a znak '.' predstavlja ugašenu lampicu).

```
###   ..#   ###   ###   #.#   ###   ###   ###   ###   ###
#.#   ..#   ..#   ..#   #.#   #..   #..   ..#   #.#   #.#
#.#   ..#   ###   ###   ###   ###   ###   ..#   ###   ###
#.#   ..#   #..   ..#   ..#   ..#   #.#   ..#   #.#   ..#
###   ..#   ###   ###   ..#   ###   ###   ..#   ###   ###
```

Kada se Amar jutros probudio osjećao se pomalo čudno i obuzela ga je nervoza da je možda propustio termin u teretani. Da nesreća bude još veća, pogledao je na sat i otkrio da upaljene lampice ne prikazuju ništa smisleno i zaključio da su neke lampice preko noći vjerojatno izgorile i više ne rade. Osjećajući jaku želju da ipak sazna da li je možda još uvijek dovoljno rano da postoji teoretska mogućnost da stigne na trening, poželio je otkriti koliko bi najmanje sati moglo biti.

Napravite program koji će na temelju izgleda sata zaključiti koje je **najranije moguće** vrijeme prikazano na satu.

### Ulazni podaci

U ulaznoj datoteci nalaze se četiri tablice znakova '#' i '.' odvojene jednim praznim stupcem koje predstavljaju izgled sata (raspored upaljenih i ugašenih lampica na satu).

### Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati traženo vrijeme.

## Test primjeri

<p><b>TRENING.IN</b></p> <pre>#.# ... .. #.. #.# ... .. #.. #.# ### ### ### #.# #.. ..# ..# ### ### ### ..#</pre> <p><b>TRENING.OUT</b></p> <p>02:34</p>	<p><b>TRENING.IN</b></p> <pre>### ### ..# ... ... #.# #.# #.# ### #.# .# .# #.. #.# ..# ..# ### #.# ..# ###</pre> <p><b>TRENING.OUT</b></p> <p>20:48</p>	<p><b>TRENING.IN</b></p> <pre>#.. #.# #.# .#. ..# #.. ... #.. #.# .## ### .## ..# #.# ... .. #.# ### #.# ###</pre> <p><b>TRENING.OUT</b></p> <p>06:25</p>
--	--	---

```
/*
  Zadatak TRENING
*/
#include <cstdlib>
#include <iostream>
#include <stdio.h>
using namespace std;
char ispravne_znamenke[10][16] = { "####.##.##.####",
    "..#.#..#.#.#",
    "###.#####.###",
    "###.#####.####",
    "#.##.#####..#.#",
    "####.###.####",
    "####.#####.####",
    "###.#.#.#.#",
    "####.#####.####",
    "####.#####.####" };
```

```

char brojke_na_satu[4][16];
struct {
    int sati,minuta;
    } rjesenje;
FILE *file;
//
void ucitaj_podatke(void)
{
    int i,j;
    //
    file = fopen("treening4.in","rt");
    for (i = 0;i < 5;++i)
        for (j = 0;j < 4;++j)
            fscanf(file,"%s",brojke_na_satu[j] + 3 * i);
    fclose(file);
}
//
int odgovara(int koja_znamenka,int broj)
{
    int i;
    //
    for (i = 0;i < 15;++i)
        if (brojke_na_satu[koja_znamenka][i] == '#' && ispravne_znamenke[broj][i] == '.')
            return 0;
    return 1;
}

```

```
//  
void rijesi(void)  
{  
    int i,j;  
    //  
    for (i = 0;i < 24;++i)  
        for (j = 0;j < 60;++j)  
            if (odgovara(0,i / 10) && odgovara(1,i % 10) &&  
                odgovara(2,j / 10) && odgovara(3,j % 10))  
                {  
                    rjesenje.sati = i;  
                    rjesenje.minuta = j;  
                    return;  
                }  
}  
//  
void ispisi_rjesenje(void)  
{  
    file = fopen("treening4.OUT","wt");  
    fprintf(file,"%02d:%02d\n",rjesenje.sati,rjesenje.minuta);  
    fclose(file);  
}  
//  
int main(void)  
{  
    ucitaj_podatke();  
    rijesi();  
}
```

```
ispisi_rjesenje();
return 0;
}
```

## Zadatak - Voćnjak

Nakon uspješne prošlogodišnje prodaje voća, vlasnik farme je odlučio da proširi svoju djelatnost i sa drugom vrstom voća. Kako vlasnik želi da njegovi potomci nasljeđe voćnjake, odlučio je da ih uči voćarstvu od malih nogu. Zato im je prošle godine dao zadatak da posade nova stabla jabuka i krušaka. On je perfekcionista, a želi tu osobinu prenijeti i na svoju decu, pa im je takođe rekao da nova stabla moraju posaditi u krug.

Djeca su uspješno obavila svoj zadatak i ove godine su posađena stabla počela rasti. Međutim, primjetio je da nešto nije u redu. Stabla jabuka i krušaka bila su izmješana! Gazda to nikako nije mogao podnijeti, pa je odlučio da presadi stabla tako da sva stabla krušaka budu jedno za drugim u krugu.

Presaćivanje vrši tako što odabere dva stabla i zamijeni im mjesta. Odredite koliko najmanje zamijena je potrebno da bi sva stabla krušaka zauzimala uzastopne pozicije u krugu.

### Ulaz

(Ulazni podaci se učitavaju sa standardnog ulaza.) U prvom redu standardnog ulaza nalazi se jedan broj  $N$  ( $1 \leq N \leq 1.000.000$ ), broj stabala u krugu. U slijedećem redu nalazi se  $N$  karaktera koji opisuju početni redoslijed stabala u krugu. Svaki karakter može da ima vrijednost ili K ili J, gdje K predstavlja stablo kruške, a J stablo jabuke.

### Izlaz

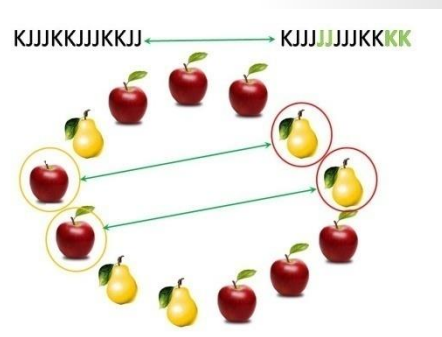
(Izlazni podaci se ispisuju na standardni izlaz.)

Prvi i jedini red standardnog izlaza treba da sadrži jedan broj - minimalan broj zamjena stabala potreban da sva stabla krušaka zauzimaju uzastopne pozicije u krugu.

Primjer 1.

```
vocnjak.in
13
KJJKKJJJKJJ
```

```
vocnjak.out
2
```





```
/*  
  Zadatak VOĆNJAK  
*/  
  
#include <cstdlib>  
#include <iostream>  
#include <string.h>  
#include <stdio.h>  
  
using namespace std;  
  
char niz[1000000];  
  
int main(int argc, char *argv[])  
{  
    freopen ( "vocnjak.in", "r", stdin );//aa  
    freopen ( "vocnjak.out", "w", stdout );//aa  
    int n, brs = 0, broj=0, p, max = 0, izlaz;  
    scanf("%d", &n);  
    memset(niz, 0, sizeof(niz));  
    scanf("%s", &niz);  
    brs = 0;  
    for(int i = 0; i < n; i++)  
    {  
        if(niz[i] == 'K')  
        {  
            brs = brs + 1;  
        }  
    }  
}
```

```

for(int i = 0; i < n; i++)
{
    broj = 0;
    p = i;
    for(int k = 0; k < brs; k++)
    {
        if(p == n)
        {
            p = 0;
        }
        if(niz[p] == 'K')
            broj++;
        p++;
    }
    if(broj >= max)
        max = broj;
}
izlaz = brs - max;
printf("%d\n", izlaz);
fclose(stdout) ;
}

```

## Zadatak -BAKTERIJE

Nakon što je ozračila bakterije radioaktivnim gama-zrakama, Naria mikroskopom promatra njihovu reakciju.

Bakterije se nalaze na kvadratnom stakalcu dimenzija  $50000 \mu\text{m} \times 50000 \mu\text{m}$ . Bakterije su oblika pravokutnika sa stranicama paralelnim rubu stakalca, a Nariji je unaprijed poznata pozicija svake bakterije.

Bakterije se međusobno ne preklapaju. Mikroskop prikazuje samo dio površine stakalca. Dio površine koji se vidi mikroskopom je dimenzija  $W \mu\text{m} \times H \mu\text{m}$  i također ima oblik pravokutnika sa stranicama paralelnim rubu stakalca.

Naria želi postaviti poziciju mikroskopa tako da na dijelu stakalca koji je vidljiv pod mikroskopom bude vidljivo što više bakterija. Napišite program koji će odrediti koliko najviše bakterija može biti vidljivo pod mikroskopom.

Napomene:

- Bakterija je vidljiva pod mikroskopom ako i samo ako je dio njene površine koji se vidi pod mikroskopom veći od 0.
- Mikroskop ne mora biti postavljen na cjelobrojnim koordinatama (vidi sliku prvog primjera test podatka).

### ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi  $W$  i  $H$  ( $1 \leq W, H \leq 50000$ ), širina i visina stakalca mikroskopa. U sljedećem retku nalazi se prirodan broj  $N$  ( $1 \leq N \leq 100000$ ), broj bakterija na stakalcu. U sljedećih  $N$  redaka, nalaze se po četiri cijela brojeva  $x_1, y_1, x_2, y_2$  ( $0 \leq x_1 < x_2 \leq 50000, 0 \leq y_1 < y_2 \leq 50000$ ), koordinate pojedine bakterije na stakalcu. Točke  $(x_1, y_1)$  i  $(x_2, y_2)$  predstavljaju suprotne vrhove pravokutnika.

### IZLAZNI PODACI

U jedini redak ispišite traženi broj bakterija.

### TEST PRIMJER 1

**ulaz**

4 3

4

2 2 5 5

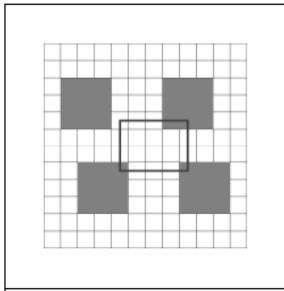
8 2 11 5

1 7 4 10

7 7 10 10

**izlaz**

3



### TEST PRIMJER 2

**ulaz**

7 4

6

5 1 8 2

2 2 4 7

5 3 8 6

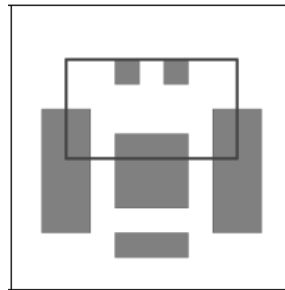
9 2 11 7

5 8 6 9

7 8 8 9

**izlaz**

5



### TEST PRIMJER 3

**ulaz**

3 3

3

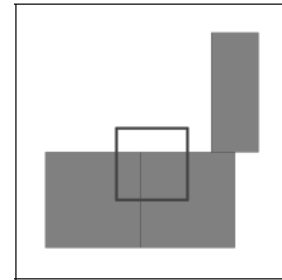
1 1 5 5

5 1 9 5

8 5 10 10

**izlaz**

2



NAPOMENA: U tablici je, za svaki test primjer, prikazana jedna od mogućih pozicija mikroskopa za rješenje.

/\*

Zadatak BAKTERIJE

\*/

```
#include <cstdio>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
using namespace std;
```

```

const int offset = 1<<17;

const int MAXN = 100010;

const int L = 100010;

int T[2*offset]; //max

int S[2*offset]; //inc za propagirat

int x1[MAXN], y1[MAXN];

int x2[MAXN], y2[MAXN];

vector<int> e[L]; //eventovi

void inc(int i, int lo, int hi, int a, int b, int k) {
    if(lo >= b || hi <= a) return;
    if(lo >= a && hi <= b) { S[i] += k, T[i] += k; return; }
    inc(i*2, lo, (lo+hi)/2, a, b, k);
    inc(i*2+1, (lo+hi)/2, hi, a, b, k);
    T[i] = max(T[i*2], T[i*2+1]) + S[i];
}

int main(void) {
    int n, w, h;
    scanf("%d %d", &w, &h);
    scanf("%d", &n);
    w = 2*w, h = 2*h;
    for(int i = 0; i < n; ++i) {
        scanf("%d %d %d %d", x1+i, y1+i, x2+i, y2+i);
        x1[i] *= 2, y1[i] *= 2, x2[i] *= 2, y2[i] *= 2;
        e[x1[i]+1].push_back(i);
        if(x2[i]+w < L) e[x2[i]+w].push_back(i);
    }
}

```

```
int ans = 0;
for(int i = 0; i < L; ++i) {
    for(int j = 0; j < (int)e[i].size(); ++j) {
        int k = (x1[e[i][j]]+1 == i ? +1 : -1); // pocetak ili kraj?
        inc(1, 0, offset, y1[e[i][j]]+1, y2[e[i][j]]+h, k);
    }
    ans = max(ans, T[1]);
}
printf("%d\n", ans);
return 0;
}
```

## REZULTATI 3. KANTONALNOG TAKMIČENJA IZ INFORMATIKE

PC^2 - board2 @ Site 1			
STOPPED			
Main Colors Tests Settings			
Rank	Name	Solved	Time
1	2 - AIRURTLE26	9	431
2	9 - BALLYCHU20	7	350
3	10 - BANAZARD101	6	433
4	3 - ANGAMON18	5	252
5	5 - ASUMON57	5	684
6	7 - AWAMON63	4	245
7	6 - ATHOZARD17	3	419
8	38 - ESTARTLE104	1	113
9	4 - ANKLEAMON82	1	148
10	22 - BUMPEZARD44	1	152
11	41 - FACEARTLE93	1	176
12	1 - AIRIRTLE89	0	0

### Rang lista

- |     |                        |                          |
|-----|------------------------|--------------------------|
| 1.  | Stjepić Ćosić Valentin | ETŠ Tuzla                |
| 2.  | Đonlagić Azur          | KŠC Tuzla                |
| 3.  | Mujačić Amir           | Gimnazija Meša Selimović |
| 4.  | Imamović Mirza         | ETŠ Tuzla                |
| 5.  | Šehović Dario          | KŠC Tuzla                |
| 6.  | Hasanbašić Amir        | ETŠ Tuzla                |
| 7.  | Kovačević Igor         | KŠC Tuzla                |
| 8.  | Maleškić Nasiha        | MSŠ Doboj Istok          |
| 9.  | Hasanbašić Sedin       | ETŠ Tuzla                |
| 10. | Vugdalić Ahmed         | Hemijska škola Tuzla     |
| 11. | Podgorčević Enver      | Gimnazija Živinice       |