BOSNA I HERCEGOVINA
FEDERACIJA BOSNE I HERCEGOVINE
TUZLANSKI KANTON
PEDAGOŠKI ZAVOD TUZLANSKOG KANTONA
TUZLA

BOSNIA AND HERZEGOVINA
FEDERATION BOSNIA AND HERZEGOVINA
TUZLA CANTON
PEDAGOGICAL INSTITUTE OF TUZLA CANTON
TUZLA

ELEKTROTEHNIČKA ŠKOLA
TUZLA

JAVNA USTANOVA MJEŠOVITA SREDNJA ELEKTROTEHNIČKA ŠKOLA TUZLA

KANTONALNO TAKMIČENJE IZ INFORMATIKE

ZA UČENIKE SREDNJIH ŠKOLA

7.

Tuzla, 30.03.2019.godine

NEŠTO O NAMA

JU Mješovita srednja Elektrotenička škola u Tuzli je osnovana 1970. godine i na samom početku se zvala Elektrotehnički školski centar. Kasnije se odvojila i postala samostalna obrazovna institucija. Tokom 49 godina svog postojanja, ova obrazovna ustanova je izgradila zavidnu reputaciju u Tuzli i u Bosni i Hercegovini.

Tokom 47 godina svoga postojanja ova Škola je stekla veliki ugled u gradu Tuzli i Državi, jer su iz nje izlazili vrlo kvalitetni stručnjaci za određena zvanja u kojima su se obrazovali. U 47 generacija maturanata, koje su izvedene, obrazovanje je završilo preko 11500 učenika. Veliki broj njih nastavilo je dalje obrazovanje na višim školama i fakultetima i to u raznim oblastima stičući različita zvanja.

Dvadesetak bivših đaka škole sada su i sami uključeni u nastavni proces kao profesori.

Javna Ustanova Mješovita srednja Elektrotehnička škola je obrazovno-vaspitna ustanova koja u svom sastavu ima Tehničku i Stručnu školu

U školskoj 2018/2019 godinu školu pohađa 704 učenika raspoređenih u 29 tehničkih i stručnih odjeljenja, a broj zaposlenih iznosi 75 od kojih 65 su nastavnici. Nastava je iz svih predmeta stručno zastupljena.

Škola je nekoliko puta posljednjih godina zauzela prvo mjesto na listi najboljih tehničkih i srodnih škola Tuzlanskog kantona. Ova škola je i pilot škola koja je u saradnji sa GTZ-om uvela nove nastavne planove i programe u sva zanimanja i koja slijedi princip ,,obrazovanje orijentirano ka djelovanju˝, što će omogućiti da učenici već u školi savladaju sva praktična znanja koja će im biti neophodno u njihovom budućem radu.

Glavni cilj ove škole je zadovoljstvo klijenata ove škole i pružanje takvog obrazovanja koje će naše učenike načiniti konkurentim na domaćem i stranom tržištu rada.

UČESNICI 7. KANTONALNOG TAKMIČENJA IZ INFORMATIKE

	NAZIV ŠKOLE	PREZIME	IME	MENTOR
1	**MSŠ BANOVIĆI**	**Kikanović**	**Adna**	Šišić Hajrija
		Rušiti	**Eldin**	Bećirović Alma
2	**MSŠ ČELIĆ**	**Mrkaljević**	**Ernad**	Suljkanović Jasmin
3	**GIMNAZIJA "dr.MUSTAFA KAMARIĆ" GRAČANICA**	**Maleškić**	**Adnan**	Husičić Nedžad
		Devedžić	**Bakir**	
4	**MSŠ GRAČANICA**	**Alibegović**	**Jasim**	Hadžić Almir
5	**GIMNAZIJA "MUSTAFA NOVALIĆ" GRADAČAC**	**Hećimović**	**Adela**	Sulejmani
		Pašalić	**Almedin**	Arijana
6	**MSŠ "HASAN KIKIĆ" GRADAČAC**	**Kujraković**	**Elmir**	Sarajlić Nazifa
		Husejnović	**Aldin**	Mujdžić Samir
		Husejnović	**Meldin**	
7	**MSŠ "MUSA ĆAZIM ĆATIĆ" KLADANJ**	**Alikadić**	**Armin**	Gazdić Senahid
8	**MS ELEKTROMAŠINSKA ŠKOLA LUKAVAC**	**Ibraković**	**Asmir**	Sakić Said
9	**MSŠ LUKAVAC**	**Kardašević**	**Ajdin**	Dženad Korman
10	**MSŠ SREBRENIK**	**Garčević**	**Amra**	Nišić Sumedin
11	**MS ELEKTROMAŠINSKA ŠKOLA TEOČAK**	**Ramić**	**Esmir**	Nihad Redžić
		Delagić	**Šeherzada**	
12	**BEHRAM BEGOVA MEDRESA**	**Halilović**	**Adis**	
		Bećirović	**Faruk**	
13	**GIMNAZIJA "ISMET MUJEZINOVIĆ"**	**Okić**	**Damir**	Hajrudin Imamović
		Jahić	**Amina**	
		Hodžić	**Mirnes**	
14	**GIMNAZIJA "MEŠA SELIMOVIĆ"**	**Šišić**	**Azra**	Muhidin Fočić
		Sukanović	**Lejla**	Mirela Šehović
		Mešić	**Kenan**	Fuad Harčin
		Malkić	**Senad**	
15	**GIMNAZIJA "SV.FRANJO"**	**Delić**	**Faris**	Bektić Mirnes

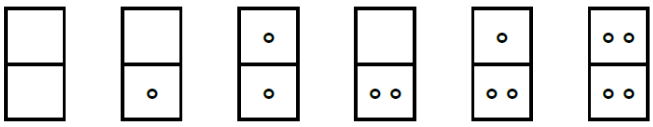
16	**MEĐUNARODNA ŠKOLA TUZLA**	**Ćidić**	**Ensar**	Delić Sadmir
		Demirović	**Amil**	Zinnur Goecken
17	**MS ELEKTROTEHNIČKA ŠKOLA**	**Smailagić**	**Sead**	Bojić Dejan
		Mujkanović	**Bakir**	Zahirović Melisa
		Mustajbašić	**Merisa**	
		Džambić	**Tarik**	
		Ahmetović	**Ajdin**	
		Sadiković	**Samir**	
		Bajrić	**Ajdin**	
18	**MS hemijska škola**	**Brkić**	**Adnan**	Bećirović Enes
		Ahmetović	**Alma**	
19	**MS MAŠINSKA ŠKOLA**	**Kadić**	**Faris**	Aščerić Amira
20	**MS SAOBRAĆAJNA ŠKOLA**	**Ikanović**	**Anida**	Klopić Danijela
21	**GIMNAZIJA ŽIVINICE**	**Šehić**	**Adel**	Puzić Halid
22	**MSŠ ŽIVINICE**	**Trumić**	**Admir**	Ilija Lučić

TAKMIČARSKI ZADACI I RJEŠENJA

Zadatak: DOMINO

Domino pločice su male pravokutne pločice koje se koriste u puno različitih igara. Na svakoj pločici nalaze se dvije oznake. Svaka oznaka sastoji se od nekog broja tačkica. Broj tačkica ovisi o veličini domino seta. U domino setu veličine **N** broj tačkica na jednoj oznaci može biti bilo koji broj između 0 i **N**, uključivo. U jednom setu ne postoje dvije domino pločice potpuno jednakih oznaka, bez obzira na redosljed oznaka na pločici. U potpunom setu veličine N se nalaze sve moguće domino pločice sa oznakama 0 do **N**.

Tako potpuni domino set veličine 2 sadrži šest pločica s oznakama:



Napišite program koji će odrediti ukupan broj tačkica na svim pločicama u potpunom domino setu veličine **N**.

ULAZNI PODACI

U prvom i jedinom retku ulaza nalazi se jedan prirodni broj, **N** ($1 \leq \mathbf{N} \leq 1000$), veličina potpunog domino seta.

IZLAZNI PODACI

U prvi i jedini redak izlaza potrebno je ispisati ukupan broj tačkica na svim pločicama u potpunom domino setu veličine **N**.

PRIMJERI TEST PODATAKA

ulaz	**ulaz**	**ulaz**
2	3	15
izlaz	**izlaz**	**izlaz**
12	30	2040

Opis drugog test primjera: Sve pločice iz seta su: [0|0], [0|1], [0|2], [0|3], [1|1], [1|2], [1|3], [2|2], [2|3] i [3|3]..

Ovaj zadatak se može rješiti čisto matematički, ovdje je prikazano više programersko rješenje.
U domino setu veličine N broj tačkica na jednoj oznaci može biti bilo koji broj između 0 i N, uključivo. U jednom setu ne postoje dvije domino pločice potpuno jednakih oznaka, bez obzira na redosljed oznaka na pločici. U potpunom setu veličine N se nalaze sve moguće domino pločice sa oznakama 0 do N.
Domino pločice u potpunom setu možemo dakle obrađivati u redosljedu počevši od pločice [0|0] pa do pločice [N|N]. Kako bi osigurali da ne obradimo neku pločicu dva puta, jednom kao [X|Y] drugi put kao [Y|X], uvodimo pravilo da obrađujemo samo pločice u kojima je prva oznaka manja ili jednaka drugoj.
Potrebno znanje:
Učitavanje broja, petlje, prepoznavanje implicitno zadanih pravilnosti

Zadatak DOMINO - Programski jezik C++

```
#include <cstdio>

int main(){
	int n;
	scanf("%d", &n);

	int sum = 0;
	for (int i = 0; i <= n; i++)
		for (int j = 0; j <= i; j++)
			sum += i + j;

	printf("%d\n", sum);
	return 0;
}
```

Zadatak: KOLONE

Poznato je da se mravi uvijek kreću u koloni. Međutim, manje je poznato što se događa kada se dvije kolone mrava sretnu u prolazu koji je preuzak da bi se dva mrava mogla mimoići. Jedna teorija kaže da u tom slučaju mravi preskaču jedni druge.
Od trenutka kada se kolone sretnu, svake sekunde svaki mrav preskoči (ili biva preskočen, kako se dogovore) mrava ispred sebe tako da ta dva mrava zamijene mjesta, ali samo ako se drugi mrav kreće u suprotnom smjeru.
Potrebno je odrediti redoslijed mrava u prolazu nakon T sekundi.

ULAZNI PODACI

U prvom redu nalaze se prirodani brojevi N1 i N2, broj mrava u prvoj i drugoj koloni.
U sljedeća dva reda nalaze se redoslijedi mrava u prvoj i drugoj koloni (od prvog prema posljednjem). Svaki mrav je jedinstveno označen velikim slovom engleske abecede (među svim mravima ne postoje dva s istom oznakom).
U zadnjem redu nalazi se cijeli broj T ($0 \leq T \leq 50$), broj sekundi proteklih od susreta kolona.

IZLAZNI PODACI

Ispišite redoslijed mrava nakon T sekundi. Naš kut gledanja je takav da nam prva kolona nailazi s lijeve, a druga kolona s desne strane.

PRIMJERI TEST PODATAKA

ulaz	**ulaz**	**ulaz**
3 3 ABC DEF 0	3 3 ABC DEF 2	3 4 JLA CRUO 3
izlaz	**izlaz**	**izlaz**
CBADEF	CDBEAF	CARLUJO

Zadatak KOLONE - Programski jezik C++

Zadatak se moze rijesiti tako da sekundu po sekundu simuliramo preskakanja (vidi rjesenje u Pascalu za implementaciju).

Efikasniji pristup je za svakog mrava izracunati na kojem ce se mjestu nalaziti.

Neka se jedan mrav nalazi na poziciji i u svojoj koloni (pozicije brojimo od 0). Mozemo izracunati njegovu poziciju P u konacnom nizu u trenutku T=0: P=N1-i-1 za prvu (lijevu) kolonu i P=N1+i za drugu (desnu) kolonu.

Razlikujemo 3 slucaja:
1. Ako je broj sekundi T manji od i, tada mrav nece preskociti nijednog mrava te ce ostati na svojoj poziciji P.
2. Ako je broj sekundi T veci od ili jednak i+(broj mrava u drugoj koloni), mrav ce preskociti sve mrave iz suprotne kolone (hoce li mu se pozicija smanjiti ili povecati ovisi o tome nalazi li se mrav s lijeve ili desne strane).
3. U preostalom slucaju pozicija mrava promijenit ce se za (T-i) (takoder ovisno o strani).

```
#include <algorithm>
#include <cstdio>

using namespace std;

int main()
{
  char k1[31], k2[31], rez[61];
  int n1, n2, t, P, i;

  scanf("%d%d", &n1, &n2);
  scanf("%s%s", k1, k2);
  scanf("%d", &t);

  for(i=0;i<n1;i++)
  {
    P = n1-i-1;
    P += min( max(t-i,0), n2 );
    rez[P] = k1[i];
  }
  for(i=0;i<n2;i++)
  {
    P = n1+i;
    P -= min( max(t-i,0), n1 );
    rez[P] = k2[i];
  }
  rez[n1+n2] = '\0';
  printf("%s\n",rez);

  return 0;
}
```

Zadatak KOLONE – Programski jezik Pascal

Zadatak se može riješiti tako da sekundu po sekundu simuliramo preskakanja. Donja implementacija pokazuje takvo rješenje.
Efikasniji pristup je za svakog mrava izravno izračunati na kojem će se mjestu nalaziti (za implementaciju vidi rješenje u jeziku C++).

```
program kolone;

var
  n1, n2, i, T, sad : integer;
  str, stanje, novo : string;
  prva, druga       : set of char;
  tmpc              : char;

begin
  readln(n1, n2);
  readln(str);
  for i:=1 to n1 do begin
    prva := prva + [ str[i] ];
    stanje := str[i] + stanje;
  end;
  readln(str);
  for i:=1 to n2 do begin
    druga := druga + [ str[i] ];
    stanje := stanje + str[i];
  end;

  readln(T);
  if T > n1+n2 then T := n1+n2;

  for sad:=1 to T do begin
    novo := stanje;

    for i:=1 to n1+n2-1 do begin
      if ( stanje[i] in prva ) and ( stanje[i+1] in druga ) then begin
        (* Skok! *)
        tmpc := novo[i];
        novo[i] := novo[i+1];
        novo[i+1] := tmpc;
      end;
    end;

    stanje := novo;
  end;

  writeln(stanje);
end.
```

Zadatak: ČOKOLADA

Mario obožava čokoladu i zato sa sobom uvijek nosi čokoladu podijeljenu na kvadratiće. Čokolada ima **R** redova i **S** stupaca. Nažalost, Mario i ne sluti da će mu njegovi prijatelji dobar dio čokolade pojesti.
Jednog dana dođe jedan prijatelj, odlomi nekoliko redova ili stupaca čokolade, pojede ih i ode. Potom dođe drugi i s preostalim komadom čokolade učini istu stvar. Na koncu dođe treći i učini istu stvar.
Pomozite Mariju i izračunajte koliko mu je koji prijatelj pojeo kvadratića čokolade.

ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi **R** i **S** ($1 \leq \mathbf{R}, \mathbf{S} \leq 10$), broj redova i broj stupaca čokolade koju je Marin nekad cijelu imao.
Svaki od sljedeća tri retka ima format "**K** redova" ili "**K** stupaca", a govori nam koliko je redaka ili stupaca od čokolade odlomio i pojeo jedan od triju prijatelja. Prirodan broj **K** bit će ispravan, tj. bit će moguće pojesti toliko redaka ili stupaca.

IZLAZNI PODACI

Za svakog od triju prijatelja u zaseban redak ispišite broj kvadratića čokolade koje je pojeo.

PRIMJERI TEST PODATAKA

ulaz	ulaz
4 4 1 stupaca 1 stupaca 1 redova	5 3 2 redova 1 stupaca 2 stupaca
izlaz	**izlaz**
4 4 2	6 3 6

Cijelo vrijeme pamtimo vrijednosti varijabli **R** i **S** koje nam predstavljaju trenutni broj redova i stupaca čokolade.
Za svako od tri unosa oblika "K redova" ili "K stupaca" valja nam učitati broj K i string A te provjeriti je li string A = "redova" ili A = "stupaca".
U prvom slučaju upravo pojedeni broj kvadratića iznosi K * S, jer svaki od K pojedenih redova čokolade trenutno sadrži S kvadratića. U drugom slučaju pojedeni broj kvadratića iznosi K * R, jer svaki od K pojedenih stupaca čokolade trenutno sadrži R kvadratića.
Dodatno, u prvom slučaju valja nam trenutni broj redova R umanjiti za K, a u drugom slučaju trenutni broj stupaca S umanjiti za K.

Potrebno znanje:
uspoređivanje stringova, naredba odlučivanja (grananja), primjena osnovnih matematičkih operacija

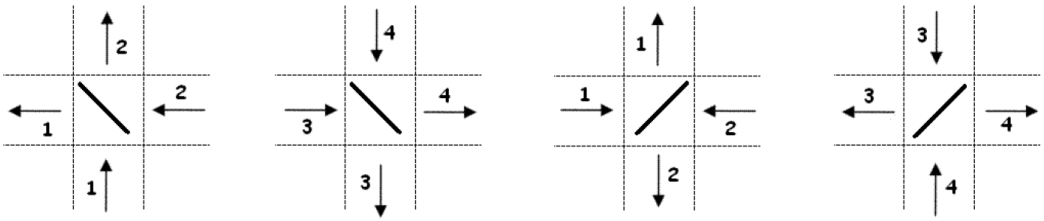
Zadatak COKOLADA - Programski jezik C++

```
#include <iostream>
using namespace std;

int main() {
	int r, s;
	cin >> r >> s;
	for (int i = 0; i < 3; ++i) {
		int k;
		string rijec;
		cin >> k >> rijec;
		if (rijec == "redova") {
			cout << k * s << endl;
			r -= k;
		}
		else {
			cout << k * r << endl;
			s -= k;
		}
	}
}
```

Zadatak: VOYAGER

Voyager 1 svemirska je sonda lansirana davne 1977. godine i u ovom trenutku polako napušta Sunčev sistem. Putujući dalje kroz svemir, u svakom sistemu planeta na koji naiđe sonda treba ostaviti poruku u obliku radio signala koja će što duže svjedočiti o njenom prolasku.
Pretpostavimo da se planetarni sistem može prikazati u obliku pravokutne mreže s **N redaka** i **M stupaca** koji prostor dijele na **N puta M jednakih polja**. U jednom polju može se nalaziti **planet**, **crna rupa** ili **prazan prostor**. Sonda je smještena u unaprijed zadanom **praznom polju** i iz njega treba u jednom od četiri moguća smjera ("U"-gore, "R"-desno, "D"-dolje, "L"-lijevo) odaslati radio signal.
Signal se nakon slanja bez prekida kreće pravolinijski kroz redak ili stupac sve dok ne stigne do planeta gdje skrene za ugao od 90 stepeni i nastavlja dalje u odgovarajućem smjeru. Postoje dvije vrste planeta koje ćemo označiti znakovima „/" i „\". Način odbijanja signala prikazan je na slici:



Signal izlazi iz sistema kad se nađe na polju u kome je crna rupa ili kada izađe izvan okvira zadane pravokutne mreže. Poznato je da signalu za prijelaz iz jednog u drugo polje treba **jedna sekunda**.
Napišite program koji će odrediti i ispisati smjer u kojem je potrebno odaslati radio signal da bi se on u sistemu zadržao **što više vremena**, kao i traženo vrijeme izraženo u sekundama. Ako signal može ostati zauvijek zarobljen u sustavu, mjesto traženog vremena ispišite poruku „Voyager".

ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi **N** ($1 \le \mathbf{N} \le 500$) i **M** ($1 \le \mathbf{M} \le 500$).
U sljedećih **N** redaka nalazi se po **M** znakova ("/" ili "\" ili "C" ili ".") pri čemu su "/" i "\" oznake planeta, "C" oznaka crne rupe, a "." oznaka praznog prostora.
U posljednjem retku nalaze se prirodni brojevi **SR** ($1 \le \mathbf{SR} \le \mathbf{N}$), broj retka, i **SS** ($1 \le \mathbf{SS} \le \mathbf{M}$), broj stupca u kojem je smještena sonda.

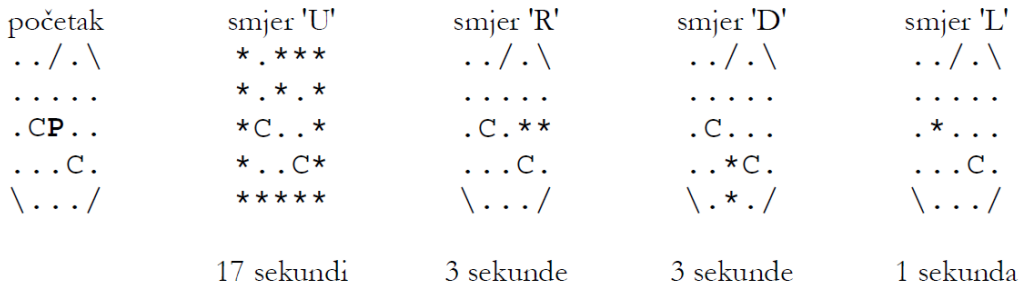
IZLAZNI PODACI

U prvi redak ispišite oznaku smjera ("U", "R", "D" ili "L"). Kada on nije jedinstven, prioritet pri odabiru ima "U", pa "R", pa "D" te na kraju "L". U drugi redak ispišite traženo vrijeme ili zadanu poruku.

TEST PRIMJERI

ulaz	ulaz	ulaz
5 5 ../.\ ..... .C... ...C. \.../ 3 3	5 5 ....\ \..\. ./\.. \../C .\../ 1 1	5 7 /.....\ ../..\. \...../ /.....\ \.\.../ 3 3
izlaz U 17	**izlaz** D 12	**izlaz** R Voyager

Pojašnjenje prvog primjera:



Rješenje zadatka svodi se na implementaciju navedenih uslova iz zadatka.

U početku, sonda se nalazi na unaprijed zadanoj poziciji. Signal se počevši od te pozicije mora poslati u sva četiri moguća smjera (gore, desno, dolje, lijevo). Znači, moramo pamtiti početnu poziciju, trenutnu poziciju i smjer kretanja. Kada signal na svom putu naiđe na planet, on treba promijeniti svoj smjer kretanja. Smjer kretanja opisujemo dvjema vrijednostima: pomakom po retku i pomakom po stupcu. Stoga kreiramo dva konstantna niza dužine 4 na sljedeći način: RP[] = {-1, 0, 1, 0} i SP[] = {0, 1, 0, -1}.
Uočimo da je vrijednostima u nizovima na nultoj poziciji opisan pomak prema gore, vrijednostima na prvoj poziciji pomak u desno, vrijednostima na drugoj poziciji pomak prema dolje te vrijednostima na trećoj poziciji pomak u lijevo. Nakon što na svakoj poziciji provjerimo vrijednost polja, trebamo napraviti korekciju smjera. Ovo ponavljamo sve dok signal ne izađe iz sistema ili dok ne naiđe na crnu rupu. Ako s **pomakr** i **pomaks** označimo trenutni smjer kretanja (na početku inicijaliziramo **pomakr** na RP[i] i **pomaks** na SP[i], za i=0..3) tada, ovisno o vrsti planeta na koji naiđemo, promjenu smjera možemo postići na sljedeći način:
// za planet s oznakom "\"
ako je (pomakr=-1) i (pomaks=0) tada { pomakr:=0; pomaks:=-1; }
inače ako je (pomakr=0) i (pomaks=-1) tada { pomakr:=-1; pomaks:=0; }
inače ako je (pomakr=0) i (pomaks=1) tada { pomakr:=1; pomaks:=0; }
inače ako je (pomakr=1) i (pomaks=0) tada { pomakr:=0; pomaks:=1; }
// za planet s oznakom "/"
ako je (pomakr=0) i (pomaks=1) tada { pomakr:=-1; pomaks:=0; }
inače ako je (pomakr=0) i (pomaks=-1) tada { pomakr:=1; pomaks:=0; }
inače ako je (pomakr=1) i (pomaks=0) tada { pomakr:=0; pomaks:=-1; }
inače ako je (pomakr=-1) i (pomaks=0) tada { pomakr:=0; pomaks:=1; }
Postoji i jednostavniji način. Trenutni smjer kretanja promatramo kao jednu vrijednost u oznaci d (d=0, gore; d=1, desno; d=2, dolje; d=3, lijevo).
Promjenu smjera postižemo korištenjem logičkog operatora ekskluzivno ILI (XOR) na bitovima. Taj operator vraća vrijednost jedan ako su vrijednosti operanada bile različite. Znajući tu činjenicu, promjenu smjera postižemo na sljedeći način:
ako je (planet = '\') tada
d ^= 3
inače ako je (planet = '/') tada
d ^= 1;
Dokaz ove tvrdnje možemo provesti iscrpljivanjem. Pretpostavimo da je trenutni smjer kretanja d=0, a planet u oznaci '\'. Binarni je zapis nule 00, a trojke 11. Novi smjer kretanja tada je vrijednost operacije 00^11=11 tj. novi je smjer 3. Lako se provjeri da je ovo istina za sve slučajeve.
Dodatni je problem određivanje situacije u kojoj signal upadne u ciklus. To ćemo postići tako da brojimo polja koja smo posjetili. Budući da poruka kroz neko polje može putovati u četiri smjera, a sistem ima **N * M** polja, postoji ukupno 4 * **N * M** položaja u kojima se poruka može nalaziti. Kada neka poruka prijeđe više polja od tog broja, očito je ponovo u nekom položaju u kojem je bila prije. Idući položaj u kojem će se poruka naći ovisi samo o njenom trenutnom položaju pa je poruka sigurno upala u ciklus.
Potrebno znanje:
dvodimenzionalno polje

Zadatak VOYAGER - Programski jezik C++

```
#include <cstdio>
#include <cstring>
using namespace std;

const int N = 502;
const int inf = 0x3f3f3f3f;

int n, m;
int r, s;
int sr, ss, di, mx, smjer;
int bio[N][N][4];
char mat[N][N];

int dr[] = { -1, 0, 1, 0 };
int ds[] = { 0, 1, 0, -1 };
char out[] = { 'U', 'R', 'D', 'L' };

int main( void ) {
	scanf( "%d %d", &n, &m );
	for( int i = 0; i < n; ++i ) scanf( "%s", mat[i] );

	scanf( "%d %d", &r, &s );
	--r; --s;

	for( int i = 0; i < 4; ++i ) {
		di = i;
		sr = r;
		ss = s;

		int uk = 0;
		while( 1 ) {
			if( sr < 0 || ss < 0 || sr >= n || ss >= m ) break;
			if( mat[sr][ss] == 'C' ) break;

			if( bio[sr][ss][di] ) {
				uk = inf;
				break;
			}
			bio[sr][ss][di] = 1;

			if( mat[sr][ss] == '/' ) {
				if( di == 0 || di == 2 ) ++di;
				else --di;
			}
			if( mat[sr][ss] == '\\' ) {
				if( di == 0 || di == 2 ) --di;
				else ++di;

				di = ( di+4 )&3;
			}

			++uk;
			sr += dr[di];
			ss += ds[di];
		}

		if( uk > mx ) {
			mx = uk;
			smjer = i;
```

```
                }
                memset( bio, 0, sizeof bio );
        }


        printf( "%c ", out[smjer] );
        if( mx == inf ) printf( "Voyager\n" );
        else printf( "%d\n", mx );

        return 0;
}
```

REZULTATI TAKMIČENJA

	Prezime	*Ime*	*Škola*	*Bodovi*	*Vrijeme*
1	Smailagić	Sead	JU MS Elektrotehnička škola Tuzla	19	1095
2	Maleškić	Adnan	JU Gimnazija "dr.M.Kamarić" Gračanica	15	1445
3	Mustajbašić	Merisa	JU MS Elektrotehnička škola Tuzla	12	659
4	Džambić	Tarik	JU MS Elektrotehnička škola Tuzla	12	765
5	Ahmetović	Ajdin	JU MS Elektrotehnička škola Tuzla	12	994
6	Devedžić	Bakir	JU Gimnazija "dr.M.Kamarić" Gračanica	12	1138
7	Sadiković	Samir	JU MS Elektrotehnička škola Tuzla	11	517
8	Alikadić	Armin	JU MSŠ "M.Ćazim Ćatić" Kladanj	11	1214
9	Malkić	Senad	JU Gimnazija "M.Selimović" Tuzla	10	443
10	Bećirović	Faruk	JU Behram Begova Medresa Tuzla	10	625
11	Šehić	Adel	JU Gimnazija Živinice	10	713
12	Mešić	Kenan	JU Gimnazija "M.Selimović" Tuzla	10	730
13	Kikanović	Adna	JU MSŠ Banovići	10	958
14	Sukanović	Lejla	JU Gimnazija "M.Selimović" Tuzla	10	1103
15	Okić	Damir	JU Gimnazija "I.Mujezinović" Tuzla	10	1348
16	Mujkanović	Bakir	JU MS Elektrotehnička škola Tuzla	9	382
17	Brkić	Adnan	JU MS Hemijska škola Tuzla	9	628
18	Halilović	Adis	JU Behram Begova Medresa Tuzla	8	337
19	Jahić	Amina	JU Gimnazija "I.Mujezinović" Tuzla	8	675
20	Šišić	Azra	JU Gimnazija "M.Selimović" Tuzla	7	250
21	Bajrić	Ajdin	JU MS Elektrotehnička škola Tuzla	7	269
22	Alibegović	Jasim	JU MSŠ Gračanica	6	608
23	Ahmetović	Alma	JU MS Hemijska škola Tuzla	2	144
24	Hodžić	Mirnes	JU Gimnazija "I.Mujezinović" Tuzla	1	92