

```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[]
        TcpClient server;
        try{
            server = new TcpC
        }catch (SocketExcep
            Console.Writeli
        }
        }
        NetworkStream ns
        int recv = ns.Re
        stringData = En
        ASCII.GetStrin
        Console.WriteLin
        while(true){
            input = Consol
            :f (input == "
                newchile
                auditin
                file
```

JU Mješovita srednja elektrotehnička škola Tuzla

VIII KANTONALNO TAKMIČENJE IZ INFORMATIKE

Tuzla, 17. maj 2022. godine

SADRŽAJ

IZVJEŠTAJ	2
UVODNA RIJEČ.....	3
JU MJEŠOVITA SREDNJA ELEKTROTEHNIČKA ŠKOLA TUZLA.....	4
OPŠTINA TUZLA	6
SPISAK UČESNIKA TAKMIČENJA.....	8
ZADACI I RJEŠENJA.....	11
Zadatak: SLOVA.....	11
Zadatak: KVADRAT	14
Zadatak: KAMIONI.....	19
Zadatak: BITCOIN	22
REZULTATI TAKMIČENJA	23

IZVJEŠTAJ

Godišnjim planom takmičenja srednjih škola, na kantonalnom nivou je predviđeno takmičenje iz informatike. Takmičenje će se održati 17. maja 2022. god. u Elektrotehničkoj školi Tuzla. Takmičenje je specifično jer zahtijeva određene tehničke i organizacijske pretpostavke.

Prema informaciji iz PPZ-a, broj učesnika je 34 iz 16 srednjih škola Tuzlanskog kantona.

Takmičenje će se organizovati po propozicijama Međunarodne olimpijade iz informatike. Takmičari rješavaju probleme u nekoliko određenih programskih jezika. Vremensko ograničenje za izradu programskih rješenja je 3 sata. Testiranje urađenih programa je automatizovano. Takmičar preko odgovarajućeg interfejsa pošalje sudiji programski kod preko instalisanog servera, koji prethodno može testirati u svom lokalnom okruženju.

Sudija kompajlira poslani programski kod i upoređuje, da li on zadovoljava predefinisane ulaze i izlaze. U slučaju da se izlaz takmičarevog programskog rješenja poklapa u potpunosti sa predefinisanim (tačnim), rješenjem takmičar dobija bodove. U slučaju da takmičar nije tačno riješio problem, on može nastaviti sa rješavanjem, s time što za svako slanje pogrešnog rješenja dobija penale na vrijeme za koje je eventualno riješio problem/e. To može uticati na konačni poredak na tabeli. Formiranje rang liste je također automatizovano.

UVODNA RIJEČ

Direktor škole: mr. sci. Sejfudin Agić, dipl.ing.el.teh.

Veliko nam je zadovoljstvo da je naša škola ovogodišnji domaćin Kantonalnog takmičenja iz informatike, koje se ujedno šesti puta organizuje u našoj školi. Cilj učenika je bio pokazati što veći nivo znanja što se moglo potvrditi i ostvarenim rezultatima. Na takmičenju je učestvovalo 34 učenika iz 16 škola sa područja Tuzlanskog kantona. Ovom prilikom čestitamo svim učenicima koji su ostvarili zavidne rezultate kao i nastavnicima uz koje su stekli pokazano znanje. Želimo učenicima i nastavnicima da nastave ostvarivati što bolje rezultate i na taj način graditi bolju budućnost za sve.

JU MJEŠOVITA SREDNJA ELEKTROTEHNIČKA ŠKOLA TUZLA

Ova škola osnovana je 1970. godine Odlukom Skupštine opština Tuzla pod nazivom Elektrotehnički školski centar Tuzla i počela je sa radom 01.09.1970.godine. Cilj osnivanja škole bio je objedinjavanje obrazovanja za sva zanimanja elektrotehničke struke u jednom školskom centru. Do 1970. godine učenici ove struke su se u Tuzli obrazovali u okviru dvije škole mješovitog tipa. Elektrotehničara jake struje i elektrotehničara slabe struje, čije je redovno školovanje trajalo 4 godine; Pogonskog električara; RTV mehaničara; elektroinstalatera; elektromontera i autoelektričara, sa trajanjem školovanja od 3 godine.

Prilikom upisa u našu Školu, koja je srednjeg stepena, učenici se opredjeljuju prema sklonostima šta će izučavati. Mogu birati:

1. TEHNIČKU ŠKOLU - četvorogodišnje školovanje
2. STRUČNU ŠKOLU - trogodišnje školovanje

U tehničkoj školi učenici se osposobljavaju za stručna zvanja:

- ✓ Tehničar računarstva
- ✓ Tehničar elektronike
- ✓ Tehničar elektroenergetike
- ✓ Tehničar mehatronike

Po završetku tehničke škole i sticanja jednog od navedenih zvanja mogu nastaviti školovanje na elektrotehničkom fakultetu ili nekom drugom od fakulteta ili tražiti odgovarajuće zaposlenje.

U stručnoj školi učenici se obrazuju za zanimanja:

- ✓ Elektroničar telekomunikacija
- ✓ Autoelektričar
- ✓ Električar (međunarodni Phare VET program)

Po završetku stručne škole ne može se izravno nastaviti školovanje na univerzitetu. Za to je potrebno izvršiti dopunsko obrazovanje. Ovi učenici se osposobljavaju za direktno zapošljavanje.

Škola je uključena u nekoliko značajnih projekata. Jedan od njih je izrada i usvajanje novih nastavnih planova i programa koji su "Orijentirani ka djelovanju" odnosno pokušavaju da prate aktuelne trendove u nauci, tehnici i tehnologiji. Uz pomoć organizacije GTZ (Njemačka organizacija za tehničku saradnju) Škola uspješno realizuje taj projekat.

Škola je redovni učesnik na Festivalima rada gdje ostvarujemo zapažene rezultate, te raznim drugim takmičenjima i festivalima unutar zemlje ali i u regionu pa i šire.

OPŠTINA TUZLA

Tuzla je jedno od najstarijih naselja u Evropi sa kontinuitetom življenja. Dokaz tome su i pronađeni ostaci starog naselja sojeničkog tipa iz vremena neolita. Arheolozi su otkrili brojna naselja sa bogatim ostacima materijalne kulture prastarih stanovnika ovog područja. Pronađen je veliki broj neolitskih glinenih posuda sa raznim ornamentima od crne, sive i crvene keramike ali i kameni noževi, sjekire, strugači i drugo. Otkriveni su ostaci hrane, ljudske kosti, životinjske kosti i razni plodovi koji su koristili za ishranu ljudi tog doba. I ovi podaci potvrdili su tvrdnje da je područje Tuzle bilo naseljeno još u mlađem kamenom dobu.

Među pronađenim arheološkim materijalima ističe se i keramička neolitska posuda čija je namjena bila kuhanje slane vode i proizvodnja soli.



Prvi pisani spomen o Tuzli potiče iz 950. godine. Te je godine bizantijski historičar i car Konstantin Porfirogenet u svom djelu "O upravljanju državom", izričito spomenuo Tuzlu kao grad, pod rimskim nazivom Salines što znači grad soli, sa napomenom da se nalazi u sastavu Raškog kneza Časlava, koji je poginuo u borbi sa Mađarima.

1463. Tuzla postaje dio Osmanskog carstva, a iz tog vremena potječe i njeno današnje ime, koje se izvodi iz turske riječi tuz, koja označava sol.



Po odlasku Osmanlija 1878. grad postaje dio Austro-Ugarske monarhije. Po završetku Prvog svjetskog rata Tuzla je, kao i cijela Bosna i Hercegovina bila dio novoosnovane Kraljevina Srba, Hrvata i Slovenaca, od 1929. zvane Kraljevina Jugoslavija.

Kada su tokom Drugog svjetskog rata 2. oktobra 1943. godine partizanske jedinice oslobodile Tuzlu od okupatora, Tuzla je bila najveći oslobođeni grad u Evropi

Po posljednjem službenom popisu stanovništva iz 1991. godine, općina Tuzla imala je 131.618 stanovnika, raspoređenih u 66 naselja. 2011 godine smatra se da Tuzla ima oko 200.000 stanovnika. Krajem rata se doselilo dosta ljudi i rodilo dosta djece i normalno je da Tuzlu gledamo kao grad novog rođenja i pune snage.

SPISAK UČESNIKA TAKMIČENJA

JU MS ELEKTROTEHNIČKA ŠKOLA TUZLA

- ✓ Hasanović Edin mentor: ZAHIROVIĆ MELISA
- ✓ Hrvić Edina mentor: PEJIĆ SLAĐANA
- ✓ Kovačević Nikola mentor: HOROZIĆ MININA
- ✓ Hadžić Emir mentor: HOROZIĆ MININA

JU MSŠ BANOVIĆI

- ✓ Kikanović Adna mentor: HOROZIĆ MININA

JU MSŠ DOBOJ ISTOK

- ✓ Alibašić Lejla mentor: DŽINIĆ RUKIJA
- ✓ Šiljić Vedad mentor: DŽINIĆ RUKIJA

JU GIMNAZIJA "Dr. Mustafa Kamarić" GRAČANICA

- ✓ Buzadžić Eldar mentor: HUSIČIĆ NEDŽAD
- ✓ Dautović Kerim mentor: HUSIČIĆ NEDŽAD
- ✓ Mujić Benjamin mentor: AHMETOVIĆ JASMIN

JU MSŠ „M.Ćazim Ćatić“ KLADANJ

- ✓ Ahmetović Harun mentor: GAZDIĆ SENAID
- ✓ Šarić Irma mentor: GAZDIĆ SENAID

JU MSŠ SREBRENİK

- ✓ Suljić Mahir mentor: NIŠIĆ DAMIR
- ✓ Salihbašić Mahir mentor: IMŠIROVIĆ NEDIM



JU MS ELEKTROMAŠINSKA ŠKOLA TEOČAK

- ✓ Mumbašić Elvin
- ✓ Nakičević Amar

mentor: JUSIĆ IZET
mentor: JUSIĆ IZET

JU BEHRAMBEGOVA MEDRESA TUZLA

- ✓ Islamović Amel
- ✓ Omerović Emira

mentor: KOZAREVIĆ ELVEDIN
mentor: KOZAREVIĆ ELVEDIN

JU GIMNAZIJA „ISMET MUJEZINOVIĆ“ TUZLA

- ✓ Smajić Harun
- ✓ Aljić Mirsad

mentor: IMAMOVIĆ HAJRUDIN
mentor: IMAMOVIĆ HAJRUDIN

JU GIMNAZIJA „MEŠA SELIMOVIĆ“ TUZLA

- ✓ Rakovac Kanita
- ✓ Hodžić Hamza
- ✓ Ibrić Ejna

mentor: ŠEHOVIĆ MIRELA
mentor: HARČIN FUAD
mentor: ŠEHOVIĆ MIRELA

JU MEĐUNARODNA ŠKOLA TUZLA

- ✓ Avdić Sara

mentor: POLUTAN NIŠIĆ HARISA

JU MS HEMIJSKA ŠKOLA TUZLA

- ✓ Knežević Sanjin
- ✓ Knežević Andrej

mentor: BEĆIROVIĆ ENES
mentor: BEĆIROVIĆ ENES

JU MS MAŠINSKA ŠKOLA TUZLA

- ✓ Muhedinović Mirza
- ✓ Teskeredžić Džemail

mentor: MEHMEDOVIĆ SAMIR
mentor: AŠČERIĆ AMIRA

JU MSŠ ŽIVINICE

✓ Smajlović Tarik

mentor: LUČIĆ ILIJA

JU GIMNAZIJA „MUSTAFA NOVALIĆ“ GRADAČAC

✓ Mulaomerović Mehmed

mentor: MUSLIMOVIĆ ALMIRA

✓ Ćorić Ajdin

mentor: MUSLIMOVIĆ ALMIRA

JU GIMNAZIJA ŽIVINICE

✓ Brčaninović Enes

mentor: ŠIŠIĆ HAJRIJA

✓ Hodžić Edin

mentor: ŠIŠIĆ HAJRIJA

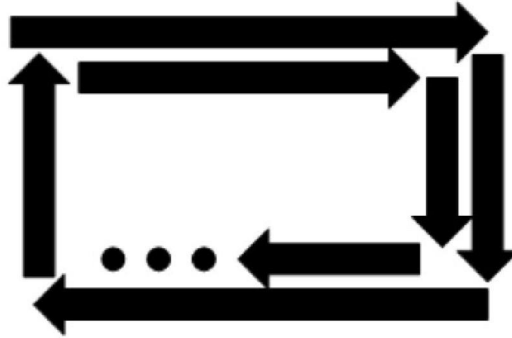
✓ Brčaninović Anel

mentor: ŠIŠIĆ HAJRIJA

ZADACI I RJEŠENJA

Zadatak: SLOVA

Zadana je matrica od $M \times N$ malih slova engleske abecede. Napišite program koji pronalazi riječ koja se dobije "spiralnim čitanjem" matrice, krećući od gornjeg lijevog polja i idući po rubu nepročitanog dijela u smjeru kazaljke sata, sve do središta matrice:



Ulazni podaci

U prvom su retku prirodni brojevi M i N ($1 \leq M, N \leq 10$) – dimenzije matrice slova, tj. broj redaka i broj stupaca.

Idućih M redaka sadrži po N malih slova engleske abecede.

Izlazni podaci

U jedini redak ispišite traženu riječ..

PRIMJERI TEST PODATAKA

```
ulaz
4 3
pro
atg
rir
ima

izlaz
programirati
```

```
ulaz
4 5
ovoje
gariz
uceja
datsi

izlaz
ovojezaistadugarijec
```

Opis prvog test primjera: vidi tekst zadatka.

Postavljamo početno polje $(1, 1)$ pa obilazimo matricu while petljom. Unutar te while petlje imamo još četiri while petlje za svaki smjer: redom za desno, dolje, lijevo i gore. Svaka od unutarnjih while petlji pomiče nas u odgovarajućem smjeru, sve dok ne dođemo do ruba matrice ili do polja koje smo već posjetili (posjećena polja označavamo točkom). Vanjska while petlja staje u trenutku kada se nijedna od unutarnjih petlji nije pomaknula ni za jedno polje. Za implementacijske detalje pogledajte priloženi kod.

Moguće rješenje u Python-u

```
m, n = map(int, input().split())
a = []
for i in range(m):
    a.append([c for c in input()])
x, y = 0, -1
while True:
    promjena = False
    # idi desno
    while y + 1 < n and a[x][y + 1] != '!':
        y += 1
        print(a[x][y], end="")
        a[x][y] = '.'
        promjena = True
    # idi dolje
    while x + 1 < m and a[x + 1][y] != '!':
        x += 1
        print(a[x][y], end="")
        a[x][y] = '.'
        promjena = True
    # idi lijevo
    while y - 1 >= 0 and a[x][y - 1] != '!':
        y -= 1
        print(a[x][y], end="")
        a[x][y] = '.'
        promjena = True
    # idi gore
    while x - 1 >= 0 and a[x - 1][y] != '!':
        x -= 1
        print(a[x][y], end="")
        a[x][y] = '.'
```

VIII Kantonalno takmičenje iz Informatike

```
    promjena = True
if not promjena:
    print()
    break
```

Zadatak: KVADRAT

Damirova najdraža igračka je magični kvadrat u kojem se nalazi 9 prirodnih brojeva poredanih u 3 retka i 3 stupca.

Magični kvadrat ima svojstvo da su zbrojevi brojeva u svakom retku, svakom stupcu i na obje dijagonale međusobno jednaki, a svaki broj u njemu je manji od ili jednak 20000 (dvadeset tisuća).

Dok je Damir spavao, lopovi su ukrali najviše tri broja iz njegovog magičnog kvadrata. Napišite program koji će za neki zadani (krnji) kvadrat odrediti brojeve koji nedostaju.

Ulazni podaci

U svakom od tri retka se nalaze tri cijela broja tj. opis magičnog kvadrata.

Na mjestima brojeva koji su ukradeni nalazi se broj 0 (nula).

Izlazni podaci

Treba ispisati tačno popunjeni magični kvadrat.

Primjeri test podataka

ulaz

```
4 9 2
3 0 7
8 1 6
```

izlaz

```
4 9 2
3 5 7
8 1 6
```

ulaz

```
0 12 12
16 10 0
8 8 14
```

izlaz

```
6 12 12
16 10 4
8 8 14
```

ulaz

```
495 468 0
0 522 414
441 0 549
```

izlaz

```
495 468 603
630 522 414
441 576 549
```

```
/* Zadatak KVADRAT
```

Ključna stvar u rješavanju zadatka je saznati kolika je magična suma (suma svakog retka, svakog stupca i glavne i sporedne dijagonale). Ako imamo redak, stupac ili dijagonalu koja je već popunjena, samo zbrojimo elemente i imamo magičnu sumu. Jedini slučaj kad nemamo niti jedan popunjeni redak, stupac ili dijagonalu, jest kad su nepoznata polja (nule) na glavnoj ili sporednoj dijagonali. Tada je magična suma jednaka polovici sume svih poznatih elemenata.

Kad imamo poznatu magičnu sumu, obrađujemo nepoznata polja jedno po jedno pazeci na to da možemo popuniti samo polje koje se nalazi u retku ili stupcu (ili dijagonali) u kojem su svi ostali elementi poznati.

```
*/
```

```
#include <stdio.h>
```

```
int main( void )
{
    int a[3][3], suma = 0;
    int i, j, ponavljanje;

    for( i = 0; i < 3; ++i )
        for( j = 0; j < 3; ++j )
            scanf("%d", &a[i][j]);

    // Postoji li popunjeni redak
    for( i = 0; i < 3; ++i )
        if( (a[i][0] == 0) + (a[i][1] == 0) + (a[i][2] == 0) == 0 )
            suma = a[i][0] + a[i][1] + a[i][2];

    // Postoji li popunjeni stupac
    for( j = 0; j < 3; ++j )
        if( (a[0][j] == 0) + (a[1][j] == 0) + (a[2][j] == 0) == 0 )
            suma = a[0][j] + a[1][j] + a[2][j];

    // Je li glavna dijagonala popunjena
    if( (a[0][0] == 0) + (a[1][1] == 0) + (a[2][2] == 0) == 0 )
        suma = a[0][0] + a[1][1] + a[2][2];

    // Je li sporedna dijagonala popunjena
    if( (a[0][2] == 0) + (a[1][1] == 0) + (a[2][0] == 0) == 0 )
        suma = a[0][2] + a[1][1] + a[2][0];

    // Ako nema popunjenog reda, stupca ili dijagonale, onda su nule
    // na dijagonali, pa je suma jednaka polovini sume svih elemenata
    if( suma == 0 )
    {
        for( i = 0; i < 3; ++i )
```



```
    for( j = 0; j < 3; ++j )
        suma += a[i][j];
    suma /= 2;
}
// Mici nule koje su same u retku ili stupcu dok ih ima
for( ponavljanje = 0; ponavljanje < 3; ++ponavljanje )
{
    for( i = 0; i < 3; ++i )
        for( j = 0; j < 3; ++j )
        {
            // Zamijenu nulu ako je sama u retku
            if( (a[i][j] == 0) && (a[i][0] == 0) + (a[i][1] == 0) + (a[i][2] == 0) == 1 )
                a[i][j] = suma - a[i][0] - a[i][1] - a[i][2];

            // Zamijenu nulu ako je sama u stupcu
            if( (a[i][j] == 0) && (a[0][j] == 0) + (a[1][j] == 0) + (a[2][j] == 0) == 1 )
                a[i][j] = suma - a[0][j] - a[1][j] - a[2][j];
        }
}
// Ispisi rjesenje.
for( i = 0; i < 3; ++i )
{
    for( j = 0; j < 3; ++j )
    {
        if( j ) printf( " " );
        printf( "%d", a[i][j] );
    }
    printf( "\n" );
}

return 0;
}
```

(*

Zadatak KVADRAT

Ključna stvar u rjesenju zadatka je saznati kolika je magična suma (suma svakog retka, svakog stupca i glavne i sporedne dijagonale). Ako imamo redak, stupac ili dijagonalu koja je već popunjena, samo zbrojimo elemente i imamo magičnu sumu. Jedini slučaj kad nemamo niti jedan popunjeni redak, stupac ili

dijagonalu, jest kad su nepoznata polja (nule) na glavnoj ili sporednoj dijagonali. Tada je magična suma jednaka polovici sume svih poznatih elemenata.

Kad imamo poznatu magičnu sumu, obradjujemo nepoznata polja jedno po jedno pazeci na to da mozemo popuniti samo polje koje se nalazi u retku ili stupcu (ili dijagonali) u kojem su svi ostali elementi poznati.

*)

program kvadrat;

```
var a : array[1..3, 1..3] of longint;
    suma, i, j, ponavljanje : longint;
begin
    // Ucitaj podatke
    for i:=1 to 3 do readln(a[i, 1], a[i, 2], a[i, 3]);

    suma:=0;
    // Postoji li popunjeni redak
    for i:=1 to 3 do
        if (a[i, 1]<>0)and(a[i, 2]<>0)and(a[i, 3]<>0) then suma:=a[i, 1]+a[i, 2]+a[i, 3];
    // Postoji li popunjeni stupac
    for i:=1 to 3 do
        if (a[1, i]<>0)and(a[2, i]<>0)and(a[3, i]<>0) then suma:=a[1, i]+a[2, i]+a[3, i];
    // Je li glavna dijagonala popunjena
    if (a[1, 1]<>0)and(a[2, 2]<>0)and(a[3, 3]<>0) then suma:=a[1, 1]+a[2, 2]+a[3, 3];
    // Je li sporedna dijagonala popunjena
    if (a[1, 3]<>0)and(a[2, 2]<>0)and(a[3, 1]<>0) then suma:=a[1, 3]+a[2, 2]+a[3, 1];
    // Ako nema popunjenog reda, stupca ili dijagonale, onda su nule
```

```
// na dijagonali, pa je suma jednaka polovini sume svih elemenata
if suma=0 then
begin
  for i:=1 to 3 do
    for j:=1 to 3 do suma:=suma+a[i, j];
  suma:=suma div 2;
end;

// Mici nule koje su same u retku ili stupcu dok ih ima
for ponavljanje:=1 to 3 do
  for i:=1 to 3 do
    for j:=1 to 3 do if a[i, j]=0 then
      begin
        // Zamijenu nulu ako je sama u retku
        if (a[i, j]=0)and(a[i, j mod 3 + 1]<>0)and(a[i, (j+1) mod 3 + 1]<>0) then
          a[i][j] := suma - a[i][1] - a[i][2] - a[i][3];

        // Zamijenu nulu ako je sama u stupcu
        if (a[i, j]=0)and(a[i mod 3 + 1, j]<>0)and(a[(i+1) mod 3 + 1, j]<>0) then
          a[i][j] := suma - a[1][j] - a[2][j] - a[3][j];
      end;

    // Ispisi rezultat
    for i:=1 to 3 do writeln(a[i, 1], ' ', a[i, 2], ' ', a[i, 3]);
end.
```

Zadatak: KAMIONI

Na parkiralište stane P automobila parkiranih jedan do drugog, svi u jednom redu. Trenutna popunjenost parkirališta zadana je nizom znakova, gdje znak '#' (ljestve) označava zauzeto parkirno mjesto, a znak '.' (tačka) označava slobodno mjesto.

Uskoro dolazi K velikih kamiona od kojih svaki zauzima tačno L uzastopnih parkirnih mjesta te je potrebno ukloniti neke parkirane automobile sa parkirališta kako bi bilo moguće parkirati sve kamione na slobodna mjesta. Kamioni ne moraju biti parkirani na susjedna mjesta.

Napišite program koji za zadano parkiralište određuje koliko je najmanje automobila potrebno ukloniti kako bi bilo moguće parkirati sve kamione.

Ulazni podaci

U prvom redu nalaze se tri prirodna broja, P , K i L ($P, K, L \leq 1000$) međusobno odvojena razmakom - broj parkirnih mjesta, broj kamiona i broj parkirnih mjesta koje svaki kamion zauzima.

U drugom redu nalazi se tačno P znakova koji predstavljaju popunjenost parkinga. Svaki znak je tačka '.' ili ljestve '#'.

Ulazni podaci će uvijek biti takvi da će biti moguće parkirati sve kamione, tj. vrijediće $K * L \leq P$

IZLAZNI PODACI

U prvi i jedini red potrebno je ispisati koliko je najmanje automobila potrebno ukloniti..

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
6 1 3	9 2 3	15 2 5
#.#.##	.##..#..#	.#.....#.#####.
izlaz	izlaz	izlaz
1	2	3

Objašnjenje 3. primjera: Ako uklonimo prva tri automobila sa lijeve strane, dobijamo 10 slobodnih mjesta na koje možemo parkirati dva kamiona dužine 5.

Jedan od načina za riješiti zadatak je iterirati po parkirnim mjestima s lijeva na desno, te na svakom mjestu isprobati dvije mogućnosti, postaviti kamion ili ne postaviti ništa i samo prijeći na iduće parkirno mjesto. Primijetite da u trenutku odlučivanja između te dvije mogućnosti nije bitno kako su raspoređeni kamioni na parkirnim mjestima prije tog trenutka, već je jedino bitno koliko je kamiona iskorišteno do sada. To znači da je moguće definirati funkciju $cijena(n, k)$ koja vraća koliko je najmanje automobila potrebno maknuti s parkinga da se uspješno rasporede k kamiona na zadnjih n parkirnih mjesta. Ta funkcija se računa isprobavajući dvije mogućnosti, postaviti ili ne postaviti kamion.

$$cijena(n, k) = \min\{cijena(n + L, k - 1) + koliko(n), cijena(n + 1, k)\},$$

gdje $koliko(n)$ označava koliko je zauzetih parkirnih mjesta u unutar sljedećih L mjesta, tj. u intervalu $[P - n, P - n + L >$.

Broj stanja ove funkcije je $O(PK)$, a prijelaz je konstante vremenske složenosti što znači da je ukupna vremenska i memorijska složenost $O(PK)$.

Primjer rješenja

```
// Kamioni

#include <iostream>

using namespace std;

const int MXP = 1001;
const int MXK = 1001;
const int INF = MXP + 1;
char parking[MXP];
int dp[MXP][MXK];
int cost[MXP][MXP]; // cost[i][j] = broj auta u intervalu [i, i+j>

int main() {
    int p, k, l;
    cin >> p >> k >> l;
    cin >> parking;

    for (int i = p - 1; i >= 0; i--) {
        if (parking[i] == '#') cost[i][1] = 1;
        else cost[i][1] = 0;
    }
    for (int j = 2; j <= l; j++) {
        for (int i = p; i > p - j; i--) {
            cost[i][j] = INF;
        }
        for (int i = p - j; i >= 0; i--) {
            cost[i][j] = cost[i + 1][j - 1];
            if (parking[i] == '#') cost[i][j]++;
        }
    }
}
```

```
    }  
    }  
    // dp[i][n] oznacava koliko je najmanje auta potrebno maknuti na dijelu  
    // parking od i-tog indeksa do kraja, a ako imamo n kamiona za sparkirati  
    dp[p][0] = 0;  
    for (int n = 1; n <= k; n++) {  
        dp[p][n] = INF;  
    }  
    for (int i = p - 1; i >= 0; i--) {  
        dp[i][0] = 0;  
        for (int n = 1; n <= k; n++) {  
            dp[i][n] = dp[i + 1][n];  
            dp[i][n] = min(dp[i][n], dp[i + 1][n - 1] + cost[i][l]);  
        }  
    }  
    cout << dp[0][k] << endl;  
    return 0;  
}
```

Zadatak: BITCOIN

Malom Denisu baka je dala 10 eura za sladoled i kolače. Denis je odlučio da ne želi jesti slatkiše nego da želi biti poduzetan pa je odlučio novac uložiti u kriptovalute. Njegov lični bankar dostavio mu je predviđanja cijene bitcoina za idućih N dana, a Denis je odlučio ulagati na sljedeći način:

- prvo će odabrati jedan dan kada će uložiti svih 10 eura u bitcoin;
- u jednom od dana nakon toga prodat će sve bitcoine za eure;
- konačno, u jednom danu od dana nakon toga opet će kupiti bitcoine (i pritom potrošiti sve eure).

Koliko je najviše bitcoina Denis može imati nakon izvršenja svoga plana?

ulazni podaci

U prvom redu se nalazi broj N ($3 \leq N \leq 100\,000$), broj dana

U svakom od sljedećih N redaka nalazi se jedan reakan broj zaokružen na 3 decimale, cijena bitcoina toga dana (u eurima). Cijena neće biti manja od 0,01 EUR ni veća od 100 000 EUR.

izlazni podaci

Ispišite jedan realan broj, najveći broj bitcoina koje Denis može imati po završetku svog plana. Rezultat će se smatrati tačnim ako odstupa od službenog za najviše 0,001.

test primjeri

ulaz

3
10.000
0.100
10.000

izlaz

0.01

ulaz

8
118.915
56.144
8.232
324.648
481.128
21.693
300.122
121.165

izlaz

26.942361

Pojašnjenje prvog primjera: Jedini način da Denis provede svoj plan je da kupuje prvi dan, prodaje drugi i opet kupuje treći dan. Prvi dan kupio je 1 bitcoin za 10€, idući dan prodao ga je za 0,1€ i treći dan od tog novca kupio 0,001 bitcoin.

Pojašnjenje prvog primjera: Kao što se vidi, tržište valuta je vrlo nepredvidljivo. Denisu se najviše isplati kupiti treći dan, prodati peti i opet kupiti šestu dan

REZULTATI TAKMIČENJA

Plasman	takmičar	bodovi	NAZIV ŠKOLE	OPĆINA
1	Islamović Amel	12	BEHRAM BEGOVA MEDRESA	TUZLA
2	Šiljić Vedad	11	MSŠ DOBOJ ISTOK	DOBOJ ISTOK
3	Suljić Mahir	11	MSŠ SREBRENİK	SREBRENİK
4	Buzadžić Eldar	10	Gimnazija "dr.M.Kamarić"	GRAČANICA
5	Hodžić Hamza	10	Gimnazija "M.Selimović"	TUZLA
6	Hasanović Edin	10	MS elektrotehnička škola	TUZLA
7	Dautović Kerim	9	Gimnazija "dr.M.Kamarić"	GRAČANICA
8	Kikanović Adna	7	MSŠ BANOVIĆI	BANOVIĆI
9	Mujić Benjamin	7	Gimnazija "dr.M.Kamarić"	GRAČANICA
10	Ahmetović Harun	7	MSŠ "M.Ćazim Ćatić" KLADANJ	KLADANJ
11	Šarić Irma	7	MSŠ "M.Ćazim Ćatić" KLADANJ	KLADANJ
12	Hadžić Emir	7	MS elektrotehnička škola	TUZLA
13	Hodžić Edin	7	Gimnazija "ŽIVINICE	ŽIVINICE
14	Alibašić Lejla	5	MSŠ DOBOJ ISTOK	DOBOJ ISTOK
14	Mulaomerović Mehmed	5	Gimnazija "M.Novalić"	GRADAČAC
14	Ćorić Ajdin	5	Gimnazija "M.Novalić"	GRADAČAC
14	Salihbašić Mahir	5	MSŠ SREBRENİK	SREBRENİK
14	Mumbašić Elvin	5	MS elektromašinska škola TEOČAK	TEOČAK
14	Nakičević Amar	5	MS elektromašinska škola TEOČAK	TEOČAK
14	Omerović Emira	5	BEHRAM BEGOVA MEDRESA	TUZLA
14	Smajić Harun	5	Gimnazija "I.Mujezinović"	TUZLA
14	Aljić Mirsad	5	Gimnazija "I.Mujezinović"	TUZLA
14	Rakovac Kanita	5	Gimnazija "M.Selimović"	TUZLA
14	Ibrić Ejna	5	Gimnazija "M.Selimović"	TUZLA
14	Sara Avdić	5	RICHMOND PARK SCHOOL	TUZLA
14	Hrvić Edina	5	MS elektrotehnička škola	TUZLA
14	Kovačević Nikola	5	MS elektrotehnička škola	TUZLA
14	Knežević Sanjin	5	MS hemijska škola	TUZLA
14	Knežević Andrej	5	MS hemijska škola	TUZLA
14	Mirza Muhedinović	5	MS mašinska škola	TUZLA
14	Mevludin Islamović	5	MS mašinska škola	TUZLA
14	Smajlović Tarik	5	MSŠ ŽIVINICE	ŽIVINICE
14	Brčaninović Anel	5	Gimnazija "ŽIVINICE	ŽIVINICE
14	Brčaninović Enes	5	Gimnazija "ŽIVINICE	ŽIVINICE