

**c0De
TK 2024**



**CODE TK
2024**

3. KOLO

**PROGRAM PODRŠKE RADU
PROGRAMERSKIH SEKCIJA**

ZDRAV SAN

U internatu, učenici imaju različite obrasce spavanja i ne postoji fiksno trajanje njihovog dnevnog sna. Tipične naučne preporuke sugerišu da je trajanje zdravog sna 8 sati dnevno. Dok započinjete svoj put kao programer, programeri su vas zadužili za vaš prvi projekat. Vaš zadatak je da napišete program koji kao ulaz uzima broj sati koje student spava dnevno i procjenjuje koliko je dobar njihov raspored spavanja. konkretno, Ako učenik spava striktno manje od 8 sati, program bi trebao ispisati MANJE. Ako student spava tačno 8 sati, program bi trebao ispisati SAVRSEN. Ako učenik spava striktno više od 8 sati, program bi trebao ispisati VIŠE. Možete li napisati takav program?

ULAZ

U jednoj liniji ulaza se nalazi cijeli broj H koji predstavlja broj sati koje učenik spava

IZLAZ

Jedna linija izlaza sadrži odgovore:

MANJE, ukoliko učenik ne spava dovoljno SAVRŠENO, ako učenik spava tačno onoliko koliko treba VIŠE, ako učenik spava više nego što je potrebno

PRIMJERI

Ulaz

2

Ulaz

13

Ulaz

8

Izlaz

MANJE

Izlaz

VIŠE

Izlaz

SAVRSEN

RJEŠENJE

```
#include <bits/stdc++.h>

using namespace std;

int main(){
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n;
    cin>>n;
    if(n == 8) cout<<"SAVRSEN";
    if(n < 8) cout<<"MANJE";
    if(n > 8) cout<<"VIŠE";

}
```

PLATFORMA ZA TAKMIČENJE

Srednjoškolski centar MK planira organizovati takmičenje u programiranju i ima namjeru kreirati platformu za takmičenje. Ova platforma ima N sudija od kojih će svaki takmičaru dati bodove od 1 do 10, pri čemu je 1 najniži broj bodova a 10 najviši. Rješenje problema se smatra dobrim ako mu svaki sudija da ocjenu veću od 4. Ako znate koliki broj bodova je dobio takmičar od svakog sudije, možete li reći da li je rješenje problema takmičara dobro ili ne ?

ULAZ

U prvom redu se nalazi broj N koji predstavlja broj sudija ($1 \leq N \leq 1000$)

U drugom redu se nalazi N cijelih brojeva razdvojenih praznim mjestom $S_1, S_2, S_3, \dots, S_n$.

IZLAZ

Ukoliko je zadovoljen kriterij iz zadatka, ispisati odgovor DA, a ukoliko nije napisati odgovor NE.

PRIMJERI

Ulaz

4
3 5 6 9

Ulaz

3
8 7 8

Ulaz

4
10 9 10 4

Izlaz

NE

Izlaz

DA

Izlaz

NE

RJEŠENJE

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

int main()
{
    int n;
    cin>>n;
    int sudija[n];
    int rez=0;
    for(int i=0; i<n; i++)
    {
        cin>>sudija[i];
        if(sudija[i]<5)
        {
            rez++;
        }
    }
    if(rez==0)
    {
        cout<<"DA";
    }
    if(rez>0)
    {
        cout<<"NE";
    }

    return 0;
}
```

NAJBOLJI IGRAČ

U igri kriket učestvuju dva tima od kojih svaki ima po 11 igrača. U tabeli sa rezultatima se nalaze poeni (runs) i wicketi (tri vertikalna štapa sa horizontalnim štapom na vrhu) koje je svaki od 22 igrača na terenu zauzeo. Da bismo odredili "najboljeg igrača", procjenjuje se učinak svakog igrača. Bodovi se dodjeljuju igraču na sljedeći način.

- Za svaki osvojeni run dobije se 1 bod
- Svaki zauzeti wicket dobije se 20 bodova

Igraču sa najvećim brojem osvojenih bodova dodjeljuje se titula "najbolji igrač". Na kraju ste dobili tabelu na kojoj su numerisani igrači brojevima od 1 do 22. pronađite "najboljeg igrača". Garantovano je da je za sve ulaze za ovaj problem "najbolji igrač", jedinstven
NAPOMENA: Igrač koji pripada timu koji je izgubio, može takođe osvojiti nagradu "najbolji igrač"

ULAZ

Na ulazu se nalaze 22 linije od kojih se u svakoj liniji nalaze po dva broja međusobno odvojena praznim mjestom.

Prvi broj predstavlja broj runova a drugi broj wicketa i-tog igrača.

IZLAZ

Na izlazu se nalazi index igrača koji ima najveći broj bodova

PRIMJERI

Ulaz

```
34 0 :: 45 0 :: 5 0 :: 85 0 :: 90 0 :: 2 2 :: 1 3 :: 0 1 :: 23 2 :: 13 1 :: 0 1 :: 34 0 :: 45 0 :: 5 0 ::  
85 0 :: 68 3 :: 2 2 :: 1 3 :: 0 1 :: 23 2 :: 13 1 :: 0 1
```

Izlaz

RJEŠENJE

```
#include <iostream>

using namespace std;

int main() {

    iosstream::sync_with_stdio(false);
    cin.tie(NULL);
    int lim=22,max=0,ind=0,t=0;
    while(lim--) {
        int a,b;
        cin>>a>>b;
        if(max<a+b*20) {
            max=a+b*20;
            ind=t+1;
        }
        t++;
    }
    cout<<ind<<'\n';
    return 0;
}
```

TRANSFORMACIJA TROŠKOVA

Amir je otišao na razgovor za posao za izvršnog direktora u finansijskom sektoru kompanije eXclusiveCost. Članovi Komisije na intervjuu su odlučili da testiraju Amirove matematičke vještine sa sljedećim problemom.

Cijeli broj se naziva dobrim ako, kada je napisan u binarnom obliku, ne sadrži broj 101 koji je podsekvencija (podniz). Na primjer:

Broj 4 je dobar jer preveden u binarni oblik je 100 i ne sadrži podsekvenciju 101. Broj 15 je također dobar jer preveden u binarni oblik 1111 također ne sadrži podsekvenciju 101

Međutim broj 9 nije dobar zato što u binarnom zapisu 1001 sadrži podsekvenciju 101. Imajte na umu da podsekvencija ne mora biti u nizu. Amiru je dat cijeli broj N na kojem može uraditi sljedeće operacije:

On može izabrati cijeli broj $x \geq 0$ i dodati $2naX$ broju N

Amirov zadatak je da pronađe minimalne troškove potrebne za transformaciju N u dobar cijeli broj. Možete li mu pomoći da pronađe ovaj minimalni trošak? Ukupni trošak niza poteza je zbir njihovih pojedinačnih troškova. Može se pokazati da je uvijek moguće transformisati N u dobar cijeli broj.

ULAZ

U jednoj liniji ulaza se nalazi cijeli broj N iz teksta zadatka..

IZLAZ

Na izlazu se nalazi minimalna transformacija troškova tj vrijednost $2naX$

PRIMJERI

Ulaz	Ulaz	Ulaz
6	10	43223
Izlaz	Izlaz	Izlaz
0	2	5929

OBJAŠNJENJE

U prvom slučaju broj 6 preveden u binarni brojni sistem je 110 i pošto nema podsekvenciju 101 to je rezultat 0

U drugom slučaju broj 10 preveden u binarni brojni sistem je 1010. On sadrži podsekvenciju 101 tako da je potrebno dodavati 2 na X kako bi dobili prvi dobar broj koji ne sadrži podsekvenciju 101. Traženi broj je 2 na $1=2$ kada 2 dodamo na 10 dobijamo 12 koji je dobar broj jer ne sadrži podsekvenciju 101.

RJEŠENJE

```
#include <bits/stdc++.h>
using namespace std;
int main(int argc, char const *argv[])
{
    int N;
    bool los = true;
    cin >> N;
    int novoN = N, n = (int)(log2(N));
    string binarni = bitset<64>(N).to_string().substr(64 - n - 1);
    while (los == true)
    {
        for (int i = 0; i < binarni.length(); i++)
        {
            if (binarni[i] == '0' && binarni[i + 1] == '1')
            {
                i = binarni.length();
            }
            else if (i == binarni.length() - 1)
            {
                los = false;
            }
        }
        if (los == true)
        {
            for (int i = binarni.length() - 1; i >= 0; i--)
            {
                if (binarni[i] == '1')
                {
                    novoN += pow(2, binarni.length() - 1 - i);
                    n = (int)(log2(novoN));
                    binarni = bitset<64>(novoN).to_string().substr(64 - n - 1);
                    i = -1; } } } }
        cout << novoN - N;
        return 0;
    }
}
```

MODUL MATRICE

Matrica B je reda $N \times N$ i zove se K-stranična ako zadovoljava sljedeće kriterije:

Svaki cijeli broj od 1 do $N \times N$ pojavljuje se tačno jednom u matrici B i

ako svaki element matrice B redukovano po modulu (ostatku) rezultujuća matrica je simetrična. Formalno to znači $B_{ij} \bmod K = B_{ji} \bmod K$ za svaki broj $1 \leq i, j \leq N$; gdje $X \bmod Y$ označava ostatak prilikom dijeljenja broja X brojem Y.

Na primjer $\begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix}$ (preloma matricu u novi red) je i 1-strana i 2-strana matrica ali nije 3-strana.

Međutim matrica $\begin{bmatrix} 1 & 5 \\ 2 & 3 \end{bmatrix}$ nije K-strana za bilo koje K, budući da sadrži ulaze koji su veći od N na $2=4$.

Matrica $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ također nije K-strana za bilo koje K jer sadrži elemente koji se ponavljaju.

Neka je M_n najveći cijeli broj za koji postoji M_n -strana matrica reda $N \times N$. Amir ima kod sebe matricu A reda $N \times N$ međutim neki od elemenata matrice nedostaju. Svaki element matrice A je ili cijeli broj čija je vrijednost između 1 i $N \times N$ ili je 0 (označava element koji nedostaje)

Pronađite broj načina kako biste nule popunili cijelim brojevima tako da je rezultujuća stranica M_n -stranična. Ovaj broj može biti veći, tako da bi odštampani modul bio 10 na 9 +7.

ULAZ

Prva linija ulaza sadrži dimenziju kvadratne matrice (broj redova i kolona) Sljedećih N redova i N kolona predstavlja matricu

IZLAZ

Na izlazu se nalazi broj načina kako biste zamijenili nule matrice A sa odgovarajućim brojevima koji će rezultirati u M_n -straničnu matricu čiji je ostatak 10 na 9 +7.

PRIMJERI

Ulaz

1

0

Ulaz

2

1 2

0 0

Ulaz

3

0 0 0

0 0 0

1 0 0

Izlaz

1

Izlaz

0

Izlaz

48

RJEŠENJE

```
#include <iostream>
#include <vector>
#include <cmath>
#include <unordered_set>

using namespace std;

int main(int argc, char** argv) {

    int n;
    cin >> n;

    long long mod = 1000000007;

    long long pairs = (n*n - n)/2;
    long long maxi = n*n - pairs;
    unordered_set<int> set;

    vector<vector<int>> arr(n, vector<int>(n));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (set.find(arr[i][j]) != set.end())
                cout << 0 << endl;
                return 0;
            }

            cin >> arr[i][j];
            if (arr[i][j] != 0) {
                set.insert(arr[i][j]);
            }
        }
    }
}
```

RJEŠENJE

```
int count = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (arr[i][j] != 0 && arr[j][i] != 0 && arr[i][j] % maxi != arr[j][i] %
            maxi) {
                cout << 0 << endl;
                return 0;
            }

            if (arr[i][j] > n*n) {
                cout << 0 << endl;
                return 0;
            }

        if (arr[i][j] == 0 && arr[j][i]) {
            arr[i][j] = arr[j][i] + maxi <= n*n ? arr[j][i] + maxi : arr[j][i] - maxi;
            count++;
        } else if (arr[j][i] == 0 && arr[i][j]) {
            arr[j][i] = arr[i][j] + maxi <= n*n ? arr[i][j] + maxi : arr[i][j] - maxi;
            count++;
        }
    }
}

unordered_set<int> set1;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (set1.find(arr[i][j]) != set1.end()) {
            cout << 0 << endl;
            return 0;
        }
    }

    if (arr[i][j] != 0) {
        set1.insert(arr[i][j]);
    }
}
}
```

RJEŠENJE

```
long long sum = 1;

    int dijag = 0;
    for (int i = 0; i < n; i++) {
        if (arr[i][i] == 0) {
            dijag++;
        }
    }

    for (int i = dijag; i > 1; i--) {
        sum *= i;
        sum %= mod;
    }

    for (int i = (pairs - count); i > 1; i--) {
        sum *= i;
        sum %= mod;
    }

    for (int i = (pairs - count); i >= 1; i--) {
        sum *= 2;
        sum %= mod;
    }

    if (sum) {
        cout << sum;
    } else {
        cout << 1;
    }

    return 0;
}A
```

REZULTATI 3. KOLA CODE TK 2024

	PREZIME	IME	ŠKOLA	MJESTO	1	2	3	4	5	SUMA
1	Feukić	Eldin	JU MS elektrotehnička škola	Tuzla	10	20	30	40	50	150
2	Aljukić	Ismar	JU Gimnazija „Meša Selimović“	Tuzla	10	20	30	40	20	120
3	Salibašić	Anel	JU Gimnazija „Meša Selimović“	Tuzla	10	20	30	40	20	120
4	Mujačić	Selma	JU Gimnazija „Meša Selimović“	Tuzla	10	20	30	40	20	120
5	Hajdić	Adnan	JU Gimnazija „dr. Mustafa Kamarić“	Gračanica	10	20	30	40	20	120
6	Čičkušić	Ismail	JU Gimnazija „Meša Selimović“	Tuzla	10	20	30	40	–	100
7	Halilović	Emir	JU MS elektrotehnička škola	Tuzla	10	20	30	8	0	68
8	Gabeljić	Emina	JU MSŠ Banovići	Banovići	10	20	30	8	–	68
9	Sudžuka	Samed	JU MS elektrotehnička škola	Tuzla	10	20	30	8	–	68
10	Aljić	Rijad	JU Gimnazija „dr. Mustafa Kamarić“	Gračanica	10	20	30	–	–	60
11	Imamović	Haris	JU MS elektrotehnička škola	Tuzla	10	20	30	–	–	60
12	Josipović	Kristina	JU MS elektrotehnička škola	Tuzla	8	16	30	–	–	54
13	Bećirović	Adem	Behram begova medresa	Tuzla	10	20	–	–	–	30
14	Omić	Kenan	JU MSŠ „Hasan Kikić“	Gradačac	6	20	–	–	–	26