

ZADATAK	FLASH	KOCKE	KAMIONI
izvorni kôd	flash.pas flash.c flash.cpp flash.cxx flash.py	kocke.pas kocke.c kocke.cpp kocke.cxx kocke.py	kamioni.pas kamioni.c kamioni.cpp kamioni.cxx kamioni.py
izvršna datoteka	flash.exe	kocke.exe	kamioni.exe
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		
broj bodova	50	70	80
	200		



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



HRVATSKI SAVEZ
INFORMATIČARA



Ministarstvo znanosti,
obrazovanja i sporta

Flash memorija je vrsta elektroničke memorije koja ne gubi informacije kada se prekine napon. Mirko je nedavno patentirao i proizveo novu vrstu jeftine memorije koju je nazvao MOR. MOR je naročito jeftina u proizvodnji, ali je složenija za rukovanje zbog svojih određenih ograničenja.

MOR memorija se sastoji od niza od M blokova, gdje se svaki blok sastoji od točno K bitova. Kod MOR memorije nije uvijek moguće postaviti pojedini bit na željenu vrijednost već su dopuštene samo sljedeće operacije:

- Pojedini bit možemo postaviti na 0 te ova operacija traje 1 milisekundu.
- Sve bitove u pojedinom bloku možemo postaviti na 1 te ova operacija traje 100 milisekundi.

Napišite program koji će za zadano početno i traženo stanje memorije pronaći **najmanje vrijeme** potrebno da se memorija iz početnog stanja dovede u traženo.

ULAZNI PODACI

U prvom redu nalaze se dva prirodna broja, M i K ($M, K \leq 20, M*K \leq 80$) međusobno odvojena razmakom - broj blokova i broj bitova u pojedinom bloku.

U drugom redu nalazi se niz od točno $M*(K+1)-1$ znakova - početno stanje memorije. U trećem redu nalazi se niz od točno $M*(K+1)-1$ znakova - traženo stanje memorije.

Početno i traženo stanje memorije su nizovi znakova koji se sastoje od M blokova međusobno odvojenih znakom ‘|’ (vertikalna crta, ASCII 124), a svaki blok se sastoji od točno K znakova ‘0’ ili ‘1’ koje predstavljaju vrijednost određenog bita u bloku.

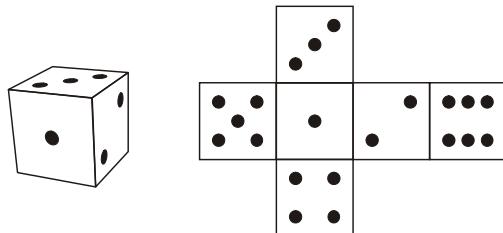
IZLAZNI PODACI

U prvi i jedini red izlaza potrebno je ispisati najmanje moguće vrijeme u milisekundama potrebno da se memorija postavi u traženo stanje.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
2 4 0110 1000 0000 0000	2 4 0110 1000 0000 0001	3 3 110 011 111 101 111 011
izlaz	izlaz	izlaz
3	105	202

Mirko i Slavko su na parkiralištu novootvorenog shopping centra na periferiji pronašli dvije ogromne kocke od stiropora. Kocke su slične onima za ‘Čovječe ne ljuti se’ (osim što su ogromne) - imaju po 6 strana označenih brojevima od 1 do 6, s brojevima raspoređenima kao na slici:



Drugim riječima, ako se kocka postavi tako da je broj jedan na prednjoj, a broj dva na desnoj strani kocke, onda će broj tri biti na gornjoj strani kocke dok će **suma brojeva na suprotnim stranama uvijek biti točno sedam**.

Mirko i Slavko su uzeli svako po jednu kocku, udaljili se dovoljno daleko, te su kocke stavili ispred sebe u istom položaju - jedan je na prednjoj, dva na desnoj, tri na gornjoj strani. Nakon toga su i Mirko i Slavko napravili N koraka, gdje je svaki korak ili prevrtanje kocke ili pauza te traje točno jednu sekundu. Točnije, mogući su sljedeći koraci:

- U - prevrtanje prema gore (eng. Up)
- D - prevrtanje prema dolje (eng. Down)
- R - prevrtanje nadesno (eng. Right)
- L - prevrtanje nalijevo (eng. Left)
- P - pauza, igrač čeka i ništa ne radi (eng. Pause)

Prevrtanja se vrše preko bridova kocke. Na primjer, ako Mirko u početnom položaju prevrne kocku prema gore, onda će ona biti u položaju u kojem je na prednjoj strani broj četiri, na desnoj strani broj dva, a na gornjoj strani broj jedan. Ako je nakon toga prevrne nadesno, onda će na prednjoj strani biti broj četiri, na desnoj broj jedan, a na gornjoj broj pet.

Mirko i Slavko su počeli u isto vrijeme te su napravili svatko po N koraka paralelno. Napišite program koji na temelju Mirkovih i Slavkovih koraka određuje **koliko je bilo koraka nakon kojih se na gornjoj strani obje kocke nalazio isti broj**.

ULAZNI PODACI

U prvom redu nalazi se prirodni broj N ($N \leq 100$) - broj koraka koje je napravio Mirko odnosno Slavko. U drugom i trećem redu nalaze se nizovi znakova M i S (svaki dužine točno N znakova) koji opisuju kako su kocku prevrtali Mirko, odnosno Slavko. U nizu se pojavljuju samo velika slova ‘U’, ‘D’, ‘R’, ‘L’, ‘P’ koja opisuju pojedine korake kako je opisano u tekstu zadatka.

IZLAZNI PODACI

U prvi red potrebno je ispisati koliko se puta dogodilo da su se nakon završenog koraka na vrhu kocaka našli isti brojevi.

PRIMJERI TEST PODATAKA

ulaz	ulaz
2 UU LL	5 UDRRP ULDDD
izlaz	izlaz
1	2

Objašnjenje 2. primjera: brojevi na vrhu kocaka podudaraju se nakon prvog te petog koraka.

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

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

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 mesta, broj kamiona i broj parkirnih mesta koje svaki kamion zauzima.

U drugom redu nalazi se točno P znakova koji predstavljaju popunjenoš parkinga. Svaki znak je točka '.' ili ljestve '#'.

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

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 automobile sa lijeve strane, dobivamo 10 slobodnih parkirnih mesta na kojih možemo parkirati dva kamiona duljine 5.