

ZADATAK	BARKOD	MEMORIJA	FRANJE
izvorni kôd	barkod.pas barkod.c barkod.cpp	memorija.pas memorija.c memorija.cpp	franje.pas franje.c franje.cpp
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		2 sekunde
memorijsko ograničenje	256 MB		
broj bodova	40	50	60
	150		



Tomislav je ljubitelj barkodova, smislenih nizova uskih i širokih vertikalnih linija koje omogućavaju elektronskoj opremi zapisivanje i očitavanje kratkih informacija. U posebnoj klasi barkodova kojom je Tomislav trenutno očaran, informacija je **niz dekadskih znamenki**, a svaka je znamenka zapisana s točno 5 uzastopnih linija koje mogu biti uske ili široke. U nastavku ćemo uske linije označavati nulom, a široke jedinicom. Prikaz znamenki dan je u sljedećoj tablici:

znamenka	kod	znamenka	kod
0	00001	5	10100
1	10001	6	01100
2	01001	7	00011
3	11000	8	10010
4	00101	9	10000

Dakle, znamenki '3' odgovara kod '11000', znamenki '8' kod '10010' itd.

Kako bi bila moguća provjera ispravnosti zapisivanja i očitavanja barkodova, na kraj svakog niza znamenaka kojeg treba kodirati najprije dodaju se dvije *zaštitne znamenke*, **C** i **K**, koje se računaju na sljedeći način:

- **C** se računa tako da se krene od prve znamenke niza (gledano s lijeva) te se redom zbroje umnošci pozicije pojedine znamenke u nizu i njene vrijednosti (prva znamenka u nizu ima poziciju 1, druga 2, itd.). Zaštitna znamenka **C** je ostatak pri djeljenu gornje sume s 10. Tako, na primjer, za niz '345' vrijednost znamenke **C** će biti ostatak od $1 \cdot 3 + 2 \cdot 4 + 3 \cdot 5$ pri djeljenu s 10 što je jednako 6. Znamenka **C** se nakon računanja dodaje na kraj niza.
- **K** se računa tako da se krene od kraja niza nastalog u prethodnom koraku (dakle niz sadrži i **netom izračunati** znak **C**), s desna na lijevo, i zbroji vrijednost svake druge znamenke (bez množenja s pozicijama!). Zaštitna znamenka **K** je sada ostatak pri dijeljenju dobivene sume s 9 te nju također dodajemo na kraj niza (desno od zaštitne znamenke **C**).

Nakon izračuna zaštitnih znamenki, sve se znamenke u nizu **zamjene** **pripadnim kodovima** i takav niz nula i jedinica tvori *barkod početnog niza*.

Ukoliko nam je dan određeni barkod, njegovu ispravnost možemo provjeriti tako da ga pretvorimo u niz znamenki, odbacimo dvije zaštitne znamenke s desna (**C** i **K**), ponovo izračunamo njihove *prave* vrijednosti, a zatim provjerimo jesu li one jednake vrijednostima za **C** i **K** koje smo na početku odbacili.

Napišite program koji vrši opisanu **provjeru zaštitnih znamenaka** zadanog niza. Zadani barkod (niz znamenaka '0' i '1') će uvijek biti duljine djeljive s 5 te će, kada se niz rastavi na grupe od po 5 binarnih znamenaka, svaka grupa odgovarati nekoj dekadskoj znamenki prema gornjoj tablici. Ukoliko su obje zaštitne znamenke **C** i **K** ispravne, potrebno je ispisati niz dekadskih znamenki koji odgovara zadanom barkodu (uključujući i znamenke **C** i **K**). U suprotnom je potrebno ispisati nazive zaštitnih znamenki koji ne odgovaraju zaštitnim znamenkama dobivenim provođenjem opisanih zaštitnih algoritama.

ULAZNI PODACI

U prvom retku ulaza nalazi se prirodni broj N ($15 \leq N \leq 1000$), broj binarnih znamenki koje čine barkod. U drugom retku ulaza nalazi se niz od N znakova '0' (nula) i '1' (jedinica) - zadani barkod čije kontrolne znamenke treba provjeriti.

IZLAZNI PODACI

Ukoliko je ulazni barkod ispravan, u prvi i jedini redak izlaza potrebno je ispisati znamenke predstavljene njime (uključujući zaštitne znamenke).

Inače, ako točno jedna od zaštitnih znamenki nije ispravna, u jedini redak izlaza potrebno je ispisati njezin naziv (veliko slovo 'C' ili 'K'), a ukoliko su kodovi za oba zaštitna znaka neispravni potrebno je ispisati velika slova 'CK'.

PRIMJERI TEST PODATAKA

ulaz

25

1100000101101000110010001

izlaz

34561

ulaz

25

1100000101101000000101100

izlaz

CK

Mirko je upravo nabavio najnoviju PC igru: *Simulator sela 2013*. Prilikom instalacije shvatio je da na svom tvrdom disku nema slobodnih M megabajta koji su potrebni za instalaciju, pa sada ubrzano pokušava osloboditi prostor kako bi što prije započeo s igrom.

Mirkov tvrdi disk organiziran je u D direktorija označenih jedinstvenim prirodnim brojevima od 1 do D te F datoteka označenih jedinstvenim prirodnim brojevima od $D+1$ do $D+F$. Za svaki direktorij i zadana je oznaka njegovog roditelja R_i (direktorija u kojem se nalazi) te vrijeme potrebno za brisanje B_i . Za svaku datoteku zadana je oznaka njenog roditelja R_i (direktorija u kojem se nalazi), vrijeme potrebno za brisanje B_i te veličina u megabajtima V_i . Direktorij s oznakom 1 je vršni direktorij i nema roditelja, što označavamo s $R_1 = 0$.

Mirko može u svakom koraku obrisati jednu datoteku ili jedan direktorij. Ako Mirko briše datoteku i , nakon B_i vremena će ona nestati te će biti oslobođeno V_i megabajta te tada Mirko opet može brisati novu datoteku ili direktorij. Za brisanje direktorija i potrebno je B_i vremena, a kada je brisanje završilo, nestaje taj direktorij zajedno sa svim direktorijima i datotekama koji se direktno ili indirektno u njemu nalaze te je broj oslobođenih megabajta prostora jednak sumi veličina svih u tom koraku obrisanih datoteka. Naravno, ako se neka datoteka ili direktorij obrišu, oni nestaju s tvrdog diska i nije ih moguće obrisati ponovo.

Napišite program koji određuje najmanje moguće vrijeme brisanja određenih direktorija i datoteka tako da se njihovim brisanjem oslobodi barem M megabajta prostora na tvrdom disku. Test podaci će biti takvi da rješenje uvijek postoji, odnosno zbroj veličina svih datoteka će biti veći ili jednak od M .

ULAZNI PODACI

U prvom retku ulaza nalaze se tri prirodna broja iz teksta zadatka odvojena razmakom, M , D i F ($1 \leq M \leq 1000$, $1 \leq D$, $F \leq 100$), broj megabajta koje je potrebno osloboditi, broj direktorija i broj datoteka na tvrdom disku.

U sljedećih D redaka ulaza nalaze se po dva prirodna broja koji opisuju direktorije, R_i i B_i ($0 \leq R_i \leq D$, $1 \leq B_i \leq 1\,000\,000$), oznaka roditelja i vrijeme potrebno za brisanje direktorija s oznakom i .

U sljedećih F redaka ulaza nalaze se po tri prirodna broja koji opisuju datoteke, R_i , B_i i V_i ($1 \leq R_i \leq D$, $1 \leq B_i \leq 1\,000\,000$, $1 \leq V_i \leq 1\,000$), oznaka roditelja, vrijeme potrebno za brisanje i veličina datoteke s oznakom $D+i$.

IZLAZNI PODACI

U prvom i jedinom retku izlaza ispišite traženo najmanje moguće ukupno vrijeme brisanja datoteka i direktorija kojim se s tvrdog diska oslobađa barem M megabajta.

BODOVANJE

U skupu test podataka vrijednom ukupno 30% bodova, vrijedit će $D = 1$, $F \leq 20$.

PRIMJERI TEST PODATAKA

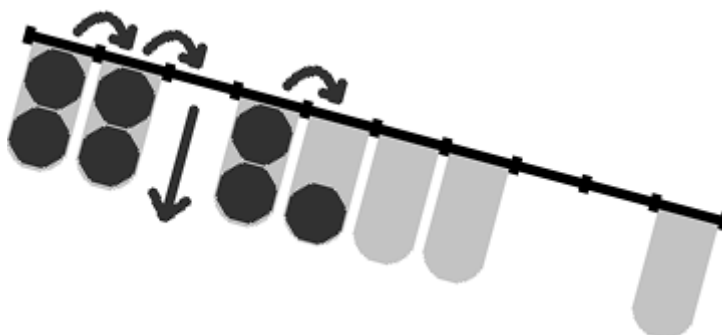
ulaz 10 1 2 0 6 1 2 5 1 3 5 izlaz 5	ulaz 10 1 3 0 6 1 3 5 1 4 6 1 5 7 izlaz 6	ulaz 10 2 1 0 8 1 9 1 7 10 izlaz 7
---	--	--

Mirku i Slavku su dosadile postojeće igračke, pa su odlučili sami konstruirati jednu novu. Skupili su nešto starih dasaka, čavala, epruveta i paket franja (staklenih kuglica u nekim krajevima Hrvatske poznatih kao “pikule” ili “špekule”), kojeg je Mirko dobio za neki od proteklih Božića. Igračku koju su sagradili možemo predstaviti kao rešetku širine 1 i dužine N . Za svaku su ćeliju u rešetki zatim pričvrstili epruvetu u koju stane točno M franja. Željeli su se brzo početi igrati pa su iz nehaja razbili K epruveta.



Poprečni presjek igračke dužine 10, s razbijenim epruvetama s oznakama 3, 8 i 9

Igra počinje! Mirko je igračku lagano nagnuo udesno, a zatim je Q puta zgrabio određenu količinu franja te ih pokušao ubaciti u neku od epruveta. Naravno, moguće je da neke franje neće stati u epruvetu u koju ih Mirko pokušava ubaciti pa će se višak “preliti” udesno. Franje će tada završiti u prvoj epruveti udesno u kojoj ima slobodnog mjesta ili će pasti kroz prvu rupu ispod koje nije epruveta. Franje se također mogu prekriti preko krajnjeg desnog ruba igračke i pasti na tlo.



Rezultat nagninjanja igračke i ubacivanja franja iznad određenih epruveta

Napišite program koji će na temelju opisa rešetke nakon svakog Mirkovog ubacivanja izračunati broj epruveta u kojima se nakon dosadašnjih ubacivanja nalazi barem jedna franja te ukupan broj franja koje su od početka igre nakon dosadašnjih ubacivanja (uključujući i trenutno) pale na tlo.

ULAZNI PODACI

U prvom retku ulaza nalaze se prirodni brojevi N i M , ($1 \leq N \leq 1\,000\,000\,000$, $1 \leq M \leq 1\,000$) duljina rešetke i kapacitet svake epruvete.

U drugom retku ulaza nalaze se prirodni broj K , broj razbijenih epruveta ($1 \leq K \leq N$, $1 \leq K \leq 100\,000$) i zatim K prirodnih brojeva E_i ($1 \leq E_i \leq N$), redni brojevi razbijenih epruveta. Epruvete su označene brojevima od 1 do N s lijeva na desno. Brojevi E_i bit će poredani uzlazno.

U trećem redu ulaznih podataka nalazi se prirodni broj Q , broj Mirkovih ubacivanja franja u epruvete ($1 \leq Q \leq 100\,000$). U svakom od sljedećih Q redova nalaze se po 2 prirodna broja, X_i i S_i ($1 \leq X_i \leq N$, $1 \leq S_i \leq 100\,000$), redni broj epruvete i količina franja koju Mirko pokušava ubaciti u i -tom ubacivanju.

IZLAZNI PODACI

Potrebno je ispisati **Q** redaka, gdje se u *i*-tom retku izlaza nalaze dva broja odvojena razmakom: broj epruveta koje nakon *i*-tog ubacivanja sadrže barem jednu franju te broj franja koje su od početka igre do tog ubacivanja (uključujući i to ubacivanje) pale na tlo.

Napomena: Rješenje ne mora nužno stati u 32-bitni cjelobrojni tip podataka.

BODOVANJE

U skupu test podataka vrijednom ukupno 20% bodova, vrijedit će $N \leq 1000$, $Q \leq 1000$.

PRIMJERI TEST PODATAKA

ulaz 10 2 3 3 8 9 3 1 5 3 7 4 3 izlaz 2 1 2 8 4 8	ulaz 12 3 4 1 4 7 11 4 3 2 2 5 2 1 8 8 izlaz 1 0 2 1 2 2 5 2
---	--

Pojašnjenje prvog test primjera: *Primjer je prikazan na slikama. U prvom ubacivanju Mirko je zagrabio pet franja te time popunio prvu i drugu epruvetu, dok je peta franja kroz rupu pala na tlo. U drugom ubacivanju Mirko je sve franje ubacio u rupu na mjestu gdje je trebala biti treća epruveta. U trećem ubacivanju popunjena je četvrta epruveta te je jedna franja završila u petoj epruveti.*