

ZADATAK	PROZORI	TETRIS	DODIR
izvorni kôd	prozori.pas prozori.c prozori.cpp prozori.cxx	tetris.pas tetris.c tetris.cpp tetris.cxx	dodir.pas dodir.c dodir.cpp dodir.cxx
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		
memorijsko ograničenje	256 MB		
broj bodova	50	70	80
	200		



Goran ima mali zaslon, a velike ambicije. U jednom trenutku shvatio je da se ne uspijeva snaći u velikom broju istovremeno otvorenih prozora te ih je odlučio rasporediti tako da se nijedna dva prozora ne preklapaju.

Kako je svaki prozor već smanjio na najmanju moguću veličinu, odlučio je koristiti jednostavan postupak. Na početku potpuno očisti zaslon, a zatim jedan po jedan prozor uzima i pokušava ga na ekranu smjestiti tako da je prozor u potpunosti unutar zaslona te se ne preklapa s nijednim ranije smještenim prozorom. Kako bi sačuvao mjesto, pokušava gornji lijevi kut prozora smjestiti na što višu poziciju, a ukoliko postoji više takvih pozicija na istoj visini, Goran odabire onu koja se nalazi najviše lijevo. Jednom kada smjesti određeni prozor, taj prozor više ne pomiče.

```
+---+  
|###|  
+---+
```

Slika 1: Prozor dimenzija 3x5

Napišite program koji će pronaći konačan raspored prozora na zaslonu. Goranov zaslon možemo zamisliti kao kvadratnu mrežu koja se sastoji od R redaka te S stupaca. Svaki jedinični kvadratić u mreži je na početku prazan, što se označava znakom '.' (točka). Dobiveni raspored prozora potrebno je prikazati na zaslonu korištenjem znakova '#', '+', '|' i '-', gdje znak '#' koristimo kako bismo prikazali unutrašnjost prozora, dok se preostala tri znaka nalaze na rubovima, kao na Slici 1. Svi prozori u zadatku sastojat će se od barem tri retka i tri stupca.

ULAZNI PODACI

U prvom redu nalaze se tri prirodna broja R , S i K ($1 \leq R, S, K \leq 30$), broj redaka i stupaca od kojih se sastoji zaslon te broj prozora koje je potrebno razmjestiti. U sljedećih K redaka nalaze se parovi prirodnih brojeva A_k , B_k ($3 \leq A_k, B_k \leq 30$), dimenzije k -tog prozora, A_k je broj redaka a B_k broj stupaca od kojih se prozor sastoji.

Prozore je potrebno razmjestiti poretком kojim su dani u ulazu, a ulazni podaci bit će takvi da je opisanim Goranovim postupkom uvijek moguće razmjestiti sve prozore.

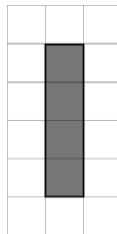
IZLAZNI PODACI

Potrebno je ispisati R redova, u svaki red točno S znakova - izgled zaslona nakon što su svi prozori razmješteni, prema uputama u tekstu zadatka.

PRIMJERI TEST PODATAKA

ulaz <pre>10 10 4 3 3 4 4 5 4 5 5</pre> izlaz <pre>+--+--+. . . # ## . . . +-+ ##+--+... +--+--+. . . # ## . . . # ## . . . # ## . . . +--+--+.</pre>	ulaz <pre>14 18 5 10 4 4 10 10 4 4 15 6 6</pre> izlaz <pre>+--+--+.+--+ # ##### ## # ##### ## # ##### ## # +-----+ ## # +-----+. . . ## # ##### . . . ## # ##### . . . ## # ##### . . . ## # ##### . . . ## # ##### . . . ## +--+--+. . .+--+ +-----+. . . ##### . . . ##### . . . +-----+. . .</pre>	ulaz <pre>20 20 16 4 5 8 4 4 7 6 4 4 7 5 4 3 5 3 8 6 7 5 4 6 5 6 4 3 6 3 8 3 4 4 3</pre> izlaz <pre>+--+--+.+--+ # ## ## ##### ## # ## ## ##### ## +--+-- ## +-----+ ## +--++. ## +-----+ ## # . ## ##### +--+ # . ## ##### . +--+ # . +--+-----+. # +--+--+. +-----+ # . . . ## ##### +--+ . . . +--+-----+. . . +-----+--+--+--+--+--+ ##### ## ##### ## ##### ## ##### ## ##### ## ##### ## ##### +--+ ## ## +-----+. . . +--+--+--+ +-----+-----+--+--+ # ## ##### ## . . . +--+--+. . .</pre>
---	---	---

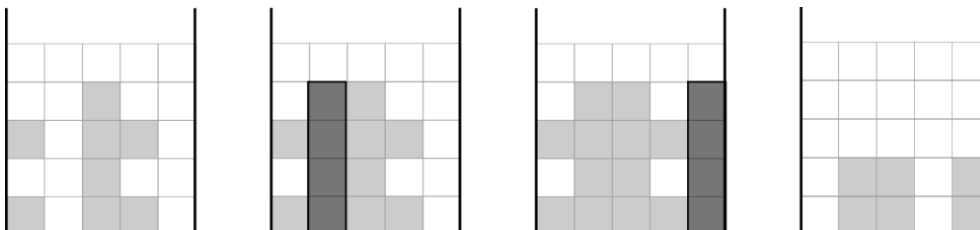
Popularna računalna igra Tetris se igra u polju koje se sastoji od jediničnih kvadratića organiziranih u 5 stupaca i nema ograničenu visinu. Svaki kvadratić polja je ili prazan ili zauzet. U pojednostavljenoj verziji igre koju ćemo razmatrati u ovom zadatku, u svakom koraku se ubacuje u polje figura koja se sastoji od četiri jedinična kvadratića poredana u jednom stupcu jedan ispod drugoga.



Slika 2: Jedina dozvoljena figura u našoj inačici igre

Figuru možemo ubaciti u bilo koji stupac. Jednom kada se odabere stupac, figura pod djelovanjem gravitacije pada sve dok se ne smjesti na dnu polja ili na već zauzetom kvadratiću. Za razliku od originalne igre, figura se ne može rotirati niti pomicati lijevo ili desno prilikom padanja.

Kada se figura smjesti odnosno završi s padanjem, tada se iz polja za igru uklanjaju svi potpuno popunjeni redci.



Slika 3: Izgled polja za igru koji odgovara prvom test primjeru dolje

Tako, na primjer, Slika 3 prikazuje tijek jedne igre. Nakon što ubacimo jednu figuru u drugi stupac te jednu figuru u peti stupac dva retka postaju potpuno popunjena te ih brišemo iz polja za igru.

Zadana je ploča za igru na kojoj su neki kvadratići već popunjeni. Napišite program koji će **pronaći neki niz koraka** koji će **očistiti** cijelu ploču osim eventualno zadnja tri redka. Drugim riječima, nakon izvođenja toga niza koraka svi kvadratići ploče koji su zauzeti se (ako postoje) moraju nalaziti u najdonja tri redka.

Pronađeni niz poteza **ne treba nužno biti minimalan** ali se ipak treba sastojati od **najviše 10000** poteza.

Još jednom, smatramo da ploča za igru ima neograničenu visinu te se, dakle, u svakom koraku figura može ubaciti u bilo koji stupac. Ulazni podaci opisuju izgled R najdonjih redaka polja za igru na početku igre, te pretpostavljamo da su tada u ostalim redcima svi kvadratići prazni. Dozvoljeno je da tijekom igre budu zauzeti kvadratići i iznad ovih zadanih R najdonjih redaka.

ULAZNI PODACI

U prvom redu nalaze se dva prirodna broja R i S ($4 \leq R, S \leq 10$), broj najdonjih redaka polja koje opisujemo te broj stupaca od kojih se polje za igru sastoji. U sljedećih R redaka nalazi se po S znakova koji opisuju početno stanje igre, gdje je znakom '.' označeno prazno polje, dok je malim slovom 'x' označeno zauzeto polje.

IZLAZNI PODACI

U prvi red potrebno je ispisati prirodan broj N manji ili jednak 10000 - broj koraka u vašem rješenju, a zatim u svaki od sljedećih N redova po jedan prirodni broj X_k ($1 \leq X_k \leq S$), redni broj stupca u koji treba ubaciti figuru u K -tom potezu.

Stupci su označeni brojevima od 1 do S s lijeva na desno.

Napomena: Rješenje ne mora biti jedinstveno.

PRIMJERI TEST PODATAKA

ulaz 4 5 ..X.. x.xx. ..X.. x.xx.	ulaz 4 7X. .x...x.X. ..x.xx.
izlaz 2 2 5	izlaz 6 1 4 7 3 5 2

Pojavom tableta i mobitela s ekranima osjetljivima na dodir, fizičke tipkovnice lagano odlaze u zaborav. Međutim, jedan problem s tipkovnicama na ekranima je veća mogućnost pogreške prilikom tipkanja, posebno kada je riječ o relativno malenim ekranima.

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

Slika 4: Raspored slova na tipkovnici

Za ovu priliku pretpostavimo da se tipkovnica sastoji od skupa pravokutnika u standardnom koordinatnom sustavu (koordinata X raste s lijeva na desno, koordinata Y raste odozdo prema gore) kao na slici gore. Tipkovnica je definirana na sljedeći način:

- Svaki pravokutnik je širine 300 te visine 200.
- Između svaka dva pravokutnika u istom 'redu' se nalazi horizontalni razmak od točno 20.
- Između dva susjedna reda je vertikalni razmak od točno 20.
- Donji lijevi kut pravokutnika koji sadrži slovo 'Q' je na koordinatama (0, 440).
- Donji lijevi kut pravokutnika koji sadrži slovo 'A' je na koordinatama (150, 220).
- Donji lijevi kut pravokutnika koji sadrži slovo 'Z' je na koordinatama (470, 0).

Na temelju ovih pravila i rasporeda tipaka danog u slici gore, mogu se odrediti koordinate pravokutnika za svako slovo. Tako su, na primjer, donji-lijevi i gornji-desni kutovi pravokutnika koji sadrži slovo 'J' jednaki (2070, 220) i (2370, 420) redom. *Središte* pravokutnika definiramo kao polovište dužine između dva suprotna kuta pravokutnika. Tako je, na primjer, središte pravokutnika koji sadrži slovo 'J' točka (2220, 320). Primijetite da kutovi svih pravokutnika imaju parne X i Y koordinate, pa stoga središte svakog pravokutnika ima cjelobrojne koordinate.

Zadan je niz od N točaka u koordinatnom sustavu - to su koordinate gdje je korisnik redom dodirnuo ekran. Za svaki dodir kažemo da je korisnik *stisnuo slovo* ako se točka nalazi **unutar** ili **na rubu odgovarajućeg pravokutnika**.

Podzadatak A. Napiši program koji će za svaku od N točaka odrediti koje je slovo korisnik stisnuo te ispisati ili nađeno slovo ili znak '?' ako točka ne odgovara niti jednom slovu.

Podzadatak B. Dodatno, zadan je rječnik koji se sastoji od M riječi, a svaka riječ od najviše 10 velikih slova engleske abecede. Vaš program treba odrediti koja je riječ iz rječnika *najbliža* zadanom nizu točaka.

Udaljenost niza točaka od riječi je **najmanja ukupna cijena niza transformacija** kojima se niz točaka može prevesti u neki niz od **točno N točaka** u kojemu se i -ta točka **nalazi unutar ili na rubu pravokutnika** i -tog slova te riječi, za svaki i od 1 do N . Pritom dozvoljene transformacije i njihove cijene su redom:

- Možemo izbrisati proizvoljnu točku iz niza točaka - cijena transformacije je zadani prirodni broj D .
- Možemo dodati proizvoljnu točku na proizvoljno mjesto u nizu - cijena je zadani prirodni broj A .
- Možemo točku iz niza (X, Y) promijeniti tj. pomaknuti **na koordinate središta pravokutnika proizvoljnog slova**. Cijena ove transformacije je kvadrat euklidske udaljenosti između točke i središta pravokutnika gdje smo točku pomaknuli. Drugim riječima, ukoliko točku (X_1, Y_1) pomaknemo na središte pravokutnika koje se nalazi na koordinatama (X_2, Y_2) onda je cijena transformacije jednaka $(X_1 - X_2)^2 + (Y_1 - Y_2)^2$. Točka ostaje na istom mjestu u nizu.

Nikakve druge transformacije niza (npr. zamjena mjesta dviju točaka) nisu dozvoljene. Dozvoljeno je da prilikom transformacija niz ostane prazan.

Ukoliko zadani niz točaka već odgovara nekoj riječi iz rječnika onda, naravno, nije potrebno raditi transformacije te je udaljenost niza točaka do te riječi jednaka nuli.

Ukoliko ima više najbližih riječi, potrebno je ispisati onu koja se **ranije pojavljuje u rječniku**.

ULAZNI PODACI

U prvom redu nalaze se četiri prirodna broja, N , M , A i D ($N \leq 10$, $M \leq 100$, $A \leq 1\,000\,000$, $D \leq 1\,000\,000$) međusobno odvojena razmakom - broj točaka, broj riječi u rječniku, cijene dodavanja i brisanja točke.

Nakon toga slijedi N redova od kojih svaki sadrži dva cijela broja X , Y ($-5000 \leq X$, $Y \leq 5000$) - koordinate odgovarajuće točke.

Nakon toga slijedi M redova, svaki sadrži jednu riječ od najviše 10 velikih slova engleske abecede. Sve riječi će biti različite i bit će poredane abecedno.

IZLAZNI PODACI

U prvi red potrebno je ispisati niz od N znakova (bez razmaka između pojedinih znakova) gdje je svaki znak ili veliko slovo engleske abecede ili znak '?' - rješenje podzadatka A.

U drugi red izlaza potrebno je ispisati jednu od riječi iz rječnika - rješenje podzadatka B.

BODOVANJE

Ukoliko je prvi red ispravan natjecatelj dobiva 40% bodova za određeni test podatak, čak i ako drugi red neispravan ili nije uopće ispisan.

Ukoliko je drugi red ispravan natjecatelj dobiva 60% bodova za određeni test podatak, čak i ako je prvi red neispravan (međutim, prvi red mora biti ispisan u ispravnom formatu).

PRIMJERI TEST PODATAKA

ulaz 4 3 100 80000 900 -8 1458 292 2181 202 3252 769 XGJP XGNO XGNP izlaz ?G?? XGNO	ulaz 00 5500 1143 436 633 541 1500 427 2058 643 1143 433 2061 55 717 511 DZEERUN EDTZRN RETURN RZETURNZ TRZZZTT ZRETURN izlaz ??????E RETURN	ulaz 4 2 100 80000 627 46 470 0 2460 100 2271 97 YZN ZYMN izlaz ZZMN ZYMN
--	---	---

Pojašnjenje drugog test primjera, podzadatak B:

Udaljenost riječi RETURN od zadanog niza točaka se dobiva sljedećim optimalnim nizom transformacija:

1. Pomicanje točke (1143, 436) u središte tipke R uz cijenu 11905.
2. Brisanje idućih 5 točaka.
3. Iduću točku nije potrebno pomicati jer je već unutar slova E.
4. Dodavanje 4 točke u središta tipaka T, U, R i N.

Ukupna cijena je niza transformacija $11905 + 5 \cdot 5500 + 4 \cdot 10500 = 81405$.

Sve riječi imaju udaljenosti redom

- RETURN 81405
- EDTZRN 85500
- ZRETURN 91905
- DZEERUN 96000
- RZETURNZ 102405
- TRZZZTT 107905

Dakle, od svih riječi iz rječnika RETURN je najbliža zadanom nizu točaka.