

Zadaci

Zadatak	PASTA	RIJEČI	POTRAGA	POLIGON
Izvršna datoteka	PASTA.EXE	RIJECL.EXE	POTRAGA.EXE	POLIGON.EXE
Izvorni kôd	PASTA.BAS PASTA.PAS PASTA.C PASTA.CPP	RIJECL.BAS RIJECL.PAS RIJECL.C RIJECL.CPP	POTRAGA.BAS POTRAGA.PAS POTRAGA.C POTRAGA.CPP	POLIGON.BAS POLIGON.PAS POLIGON.C POLIGON.CPP
Ulazna datoteka	PASTA.IN	RIJECL.IN	POTRAGA.IN	POLIGON.IN
Izlazna datoteka	PASTA.OUT	RIJECL.OUT	POTRAGA.OUT	POLIGON.OUT
Vremensko ograničenje (po test podatku)	10 sekundi	10 sekundi	10 sekundi	10 sekundi
Broj test podataka	10	10	10	10
Broj bodova (po test podatku)	3	4	6	7
Ukupno bodova	30	40	60	70
	200			

PASTA

Dokument u tekst procesoru sastoji se od N linija teksta. U prvoj liniji nalazi se zapisan broj 1, u drugoj liniji broj 2, i tako redom do N -te linije u kojoj je zapisan broj N .

Nad tim linijama izvršeno je M operacija 'cut' i 'paste'. Svaka se od tih operacija sastoji od odabira neke skupine uzastopnih linija koja se **uklanja** ('cut') iz teksta i potom **ubacuje** ('paste') na neko mjesto u novom tekstu.

Napišite program koji će za zadani niz operacija 'cut' i 'paste' odrediti sadržaj **prvih 10 linija** teksta nakon izvršenja tih operacija.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva prirodna broja N i K međusobno odvojena jednim razmakom, $10 \leq N \leq 100000$ (sto tisuća), $1 \leq K \leq 1000$ (tisuću), broj linija teksta u dokumentu i broj operacija 'cut' i 'paste'.

Svaki od sljedećih K redaka sadrži podatke o jednoj 'cut' i 'paste' operaciji i to redom izvršavanja tih operacija. U svakom od redaka nalaze se po tri prirodna broja A , B i C međusobno odvojena s po jednim razmakom, $1 \leq A \leq B \leq N$, $0 \leq C \leq N-(B-A+1)$. Broj A predstavlja redni broj prve selektirane linije, broj B redni broj zadnje selektirane linije, a broj C redni broj linije nakon izvršenja operacije 'cut' **iza** koje treba ubaciti selektiranu grupu linija. Ako je broj C jednak nuli, tada selektiranu grupu linija treba ubaciti na početak teksta.

Izlazni podaci

Izlazna datoteka sastoji se od 10 redaka u kojima se redom nalaze brojevi zapisani u prvih 10 linija teksta nakon izvršenih operacija.

Test primjeri

PASTA.IN

15 1
1 15 0

PASTA.OUT

1
2
3
4
5
6
7
8
9
10

PASTA.IN

13 3
6 12 1
2 9 0
10 13 8

PASTA.OUT

6
7
8
9
10
11
12
2
3
4

PASTA.IN

1000 6
3 7 4
1 100 57
50 60 200
63 70 500
1 800 4
7 77 98

PASTA.OUT

801
802
803
804
101
102
36
37
38
39

RIJEČI

Io i Ao igraju igru riječima. Naizmjenice govore riječi **sastavljene samo od samoglasnika** tako da **prvo slovo** svake nove riječi **odgovara zadnjem slovu** prethodno izgovorene riječi. Igra može početi izgovaranjem bilo koje riječi.

Tijekom igre zabranjeno je ponavljati već izgovorene riječi, a kako bi se smanjila mogućnost varanja dozvoljeno je koristiti samo riječi iz unaprijed zadanog rječnika.

Težina igre se definira kao zbroj duljina svih riječi koje su izgovorene tijekom igre.

Napišite program koji će na temelju rječnika odrediti kolika je **najveća moguća** težina igre koja se može odigrati birajući riječi iz zadanog rječnika.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se prirodni broj N , $1 \leq N \leq 16$, broj riječi u rječniku.

Svaki od sljedećih N redaka sadrži jednu riječ iz rječnika. Riječ je niz znakova maksimalne duljine 100 znakova, a jedini znakovi koji će se pojavljivati u riječi će biti **A, E, I, O** i **U**.

Sve riječi će biti različite.

Izlazni podaci

U prvi i jedini redak izlazne datoteke potrebno je zapisati traženu najveću moguću težinu igre.

Test primjeri

RIJECI.IN

3
AEIOU
UIU
EO

RIJECI.OUT

8

RIJECI.IN

4
AEEEO
OEOAEEO
AO
O

RIJECI.OUT

13

RIJECI.IN

5
IOO
IUUO
AI
OIOOI
AOOI

RIJECI.OUT

16

POTRAGA

Mali Živčo je delikvent. Nakon što je peti put pobjegao iz popravnog doma, ukrao je crnu limuzinu i odvezao se u grad da se malo proveseli. Međutim, Živčo nije znao da automobil koji je ukrao pripada policiji i da u njemu postoji senzor koji javlja podatke o kretanju automobila.

Senzor u automobilu je malo zastario pa ne javlja poziciju niti brzinu automobila, nego samo **smjer** u kojem se automobil kreće.

Pomozite policiji da izvede malog Živču na pravi put! Napišite program koji će na temelju mape grada, početne pozicije automobila i niza smjerova u kojima se automobil kretao odrediti **sve** pozicije na mapi na kojima se automobil može nalaziti **na kraju** Živčinog putovanja.

Mapa grada je zadana kao pravokutna ploča znakova, znak **.** (točka) označava polje po kojem se automobil može kretati dok znak **X** označava neprohodno polje. Pozicija na kojoj se nalazio Živčo na početku svog putovanja označena je znakom *****. Ta pozicija se također smatra **prohodnim** poljem.

Automobil se može kretati u četiri smjera: prema sjeveru (gore), jugu (dolje), zapadu (lijevo) i istoku (desno). Opis Živčinog puta je niz smjerova u kojima se vozio. U svakom navedenom smjeru se Živčo pomaknuo za **jedno ili više** prohodnih polja.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva prirodna broja R i S međusobno odvojena jednim razmakom, $1 \leq R \leq 50$, $1 \leq S \leq 50$, broj redaka i broj stupaca mape grada.

U svakom od sljedećih R redaka nalazi se niz od S znakova, izgled mape (znakovi **.** (točka), **X** i *****).

U sljedećem retku nalazi se prirodni broj N , $1 \leq N \leq 1000$, broj smjerova.

U svakom od sljedećih N redaka nalazi se niz znakova **SJEVER**, **JUG**, **ZAPAD** ili **ISTOK**, smjer u kojem se kretao auto.

Niti jedna dva uzastopna smjera nisu jednaka.

POTRAGA

Izlazni podaci

U svaki od R redaka izlazne datoteke treba zapisati niz od S znakova koji predstavlja mapu grada na kojoj su označene tražene pozicije pri čemu znak ***** treba pisati **samo** na onim mjestima na kojima bi se mogao nalaziti automobil **na kraju** putovanja.

Test primjeri

POTRAGA.IN

```
3 4
....
*..X
X.X.
2
ISTOK
SJEVER
```

POTRAGA.OUT

```
.*.
...X
X.X.
```

POTRAGA.IN

```
4 5
.....
.X...
...*X
X.X..
3
SJEVER
ZAPAD
JUG
```

POTRAGA.OUT

```
.....
*X*..
*..X
X.X..
```

POTRAGA.IN

```
10 9
.....X
X..XX..X.
.X.XX.X..
...XX....
...XX....
.XXX..XX.
.....X.
..XXX.X..
X.X....X.
*.....X..
4
ISTOK
SJEVER
ISTOK
JUG
```

POTRAGA.OUT

```
.....X
X..XX.*X.
.X.XX.X..
...XX....
...XX***
.XXX..XX*
.....X*
..XXX*X.*
X.X..*.X*
....**X.*
```

POLIGON

U ravnini je zadano N točaka s pozitivnim cijelobrojnim koordinatama.

Konveksni poligon kojeg možemo dobiti tako da za jedan njegov vrh uzmemo **ishodište** koordinatnog sustava, a za ostale vrhove neke od zadanih točaka i to tako da je broj vrhova tog poligona **najveći mogući** zovemo **konveksni poligon s maksimalnim brojem vrhova**.

Napišite program koji će odrediti **broj vrhova** u takvom poligonu.

Poligon je konveksan ako za svake dvije točke unutar njega, dužina koja ih spaja također leži unutar poligona.

Niti jedna dva uzastopna brida poligona **ne smiju** biti paralelna.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se prirodni broj N , $2 \leq N \leq 100$, broj zadanih točaka.

U svakom od sljedećih N redaka nalaze se prirodni brojevi X i Y međusobno odvojeni jednim razmakom, $1 \leq X \leq 100$, $1 \leq Y \leq 100$, koordinate zadanih točaka. Sve točke će biti različite.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati broj vrhova traženog poligona.

Napomena: rješenje će uvijek biti barem 3.

Test primjeri

POLIGON.IN	POLIGON.IN	POLIGON.IN
5	8	10
4 2	10 8	9 6
2 2	3 9	1 7
2 3	2 8	2 2
3 2	2 3	3 9
3 1	9 2	8 7
	9 10	3 2
POLIGON.OUT	10 3	9 4
4	8 10	3 1
		9 7
	POLIGON.OUT	6 9
	8	POLIGON.OUT
		7

Pojašnjenje za test primjer #2 (vrhovi poligona)

2 8
3 9
8 10
9 10
10 8
10 3
9 2
0 0