

Zadaci

Zadatak	DEKOD	POLICE	KALK
Izvršna datoteka	DEKOD.EXE	POLICE.EXE	KALK.EXE
Izvorni kôd	DEKOD.BAS DEKOD.PAS DEKOD.C DEKOD.CPP	POLICE.BAS POLICE.PAS POLICE.C POLICE.CPP	KALK.BAS KALK.PAS KALK.C KALK.CPP
Ulazna datoteka	DEKOD.IN	POLICE.IN	KALK.IN
Izlazna datoteka	DEKOD.OUT	POLICE.OUT	KALK.OUT
Vremensko ograničenje (po test podatku)	10 sekundi	10 sekundi	10 sekundi
Broj test podataka	10	10	10
Broj bodova (po test podatku)	3	5	7
Ukupno bodova	30	50	70
	150		

DEKOD

Jedan od načina na koji možemo šifrirati neki tekst je da svako slovo u tekstu zamijenimo nekim drugim slovom pomoću tablice zamjena koju kreiramo na sljedeći način:

Prvo odaberemo jednu riječ koja se sastoji od međusobno različitih slova engleske abecede i nazovemo je **ključna riječ**. Zatim odaberemo i jedan prirodni broj **K** manji ili jednak od 26 i nazovemo ga **ključni broj**.

Sada kreiramo tablicu po kojoj ćemo zamijenjivati slova prilikom šifriranja. Tablica se sastoji od 2 retka i 26 stupaca. U gornji redak abecednim redom upisujemo slova engleske abecede. U donjem retku krenemo od K-tog stupca i slijeva na desno upisujemo ključnu riječ slovo po slovo. Kad upišemo cijelu ključnu riječ, od sljedećeg stupca upisujemo abecednim redom slova koja se nisu pojavila u ključnoj riječi. Kada prilikom upisivanja ključne riječi ili ostalih slova stignemo do zadnjeg stupca u tablici, tada dalje nastavljamo od prvog stupca.

Npr. ako je ključna riječ DUBROVNIK, a ključni broj 10, tablica izgleda ovako:

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

Originalni tekst se šifrira tako da se svako slovo zamijeni nekim drugim slovom prema tablici zamjena. Pojedino slovo iz prvog retka tablice zamjenjuje se slovom zapisanim ispod njega.

Napišite program koji će na temelju ključne riječi i ključnog broja iz šifriranog teksta generirati **originalni tekst**.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se ključna riječ. Jedini dozvoljeni znakovi su velika slova engleske abecede (A-Z), a broj znakova u ključnoj riječi bit će manji ili jednak od 26.

U drugom retku nalazi se prirodni broj K, $1 \leq K \leq 26$, ključni broj.

U trećem retku nalazi se šifrirani tekst. Jedini dozvoljeni znakovi su velika slova engleske abecede (A-Z), a broj znakova u šifriranom tekstu bit će manji ili jednak od 100.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati originalni tekst.

Test primjeri

DEKOD.IN

NOVI
15
DTZNMNXAWT

DEKOD.OUT

VINODOLSKI

DEKOD.IN

DUBROVNIK
10
SVPKVSABZOMSRZY

DEKOD.OUT

DOBRODOSLINADMIH

DEKOD.IN

ZAGREB
23
QYELREDEWEMLFNEIEP

DEKOD.OUT

OVAJZADATAKJELAGAN

POLICE

Ormar u policijskoj postaji sastoji se od pretinaca. Pretinci su organizirani u S stupaca i R redova. Da bi se dohvatile stvari s bilo kojeg pretinca, potrebno je koristiti ljestve.

Ljestve se mogu postaviti uz određeni stupac pretinaca. Penjanjem po ljestvama na određenu visinu mogu se uzeti sve stvari iz pretinaca **do te visine** iz tog stupca, te iz lijevog i desnog susjednog stupca.

Policijci trebaju pokupiti stvari iz određenih pretinaca. Da bi se smanjio rizik ozljede na poslu, potrebno je naći način da se pokupe sve stvari, a da ukupna visina penjanja bude **najmanja moguća**. **Ukupna visina** je zbroj visina svih učinjenih penjanja.

Napišite program koji će za zadani skup pretinaca odrediti kolika je najmanja ukupna visina penjanja potrebna da bi se pokupile stvari s tih pretinaca.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva prirodna broja S i R međusobno odvojena jednim razmakom, koji predstavljaju broj stupaca i broj redaka, $1 \leq S \leq 100$, $1 \leq R \leq 100$.

U drugom retku datoteke se nalazi prirodni broj N , $1 \leq N \leq 100$, broj pretinaca koje treba dohvatiti.

U svakom od sljedećih N redaka se nalaze po dva prirodna broja A i B međusobno odvojena jednim razmakom, $1 \leq A \leq S$, $1 \leq B \leq R$, brojevi stupca i retka pretinaca koje treba dohvatiti.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati najmanju ukupnu visinu potrebnog penjanja.

Test primjeri

POLICE.IN	POLICE.IN	POLICE.IN
5 5	6 20	10 10
3	4	5
2 3	5 6	9 1
3 4	1 1	7 6
4 4	6 1	5 8
	3 7	4 1
POLICE.OUT	POLICE.OUT	POLICE.OUT
4	9	11

KALK

Mirko ima simbolički kalkulator koji ima memoriju u obliku liste čiji su elementi aritmetički izrazi i koja može sadržavati **najviše 100 izraza**. Naredbe koje kalkulator razumije su sljedeće:

HASH – ako se na prvom mjestu liste nalazi izraz **s**, a na drugom izraz **t**, onda ova naredba briše ta dva elementa iz liste i na prvo mjesto liste dodaje izraz **(t#s)**

DOLLAR – ako se na prvom mjestu liste nalazi izraz **s**, a na drugom izraz **t**, onda ova naredba briše ta dva elementa iz liste i na prvo mjesto liste dodaje izraz **(t\$s)**

SWAP – međusobno zamjenjuje prvi i drugi izraz u listi

DROP – briše prvi izraz iz liste

DUP – udvostručuje prvi izraz liste, tj. njegovu kopiju dodaje na prvo mjesto

ROT – rotira prva četiri izraza u listi tako da prvi izraz postaje drugi, drugi izraz postaje treći, treći izraz postaje četvrti, a četvrti izraz postaje prvi

Potrebno je napisati program koji će generirati niz naredbi za kalkulator takav da ako se na početku izvođenja u listi nalaze samo A i B (A na drugom mjestu, B na prvom mjestu), sve naredbe se izvedu bez greške te se na kraju izvođenja kao **jedini** element u listi **nalazi zadani izraz**.

Do greške prilikom izvođenja naredbe može doći ako se lista prepuni ili ako u listi nema dovoljno argumenata za naredbu. Za naredbe DUP i DROP potreban je barem jedan element u listi, za naredbe HASH, DOLLAR i SWAP barem dva, dok je za naredbu ROT potrebno barem četiri elementa.

Vaš program treba generirati **bilo koji** niz naredbi koji računa zadani izraz; ipak broj naredbi treba biti **manji ili jednak** od 10,000.

Ulazni podaci

U prvom i jedinom retku ulazne datoteke nalazi se zadani izraz. Izraz se sastoji od znakova **A**, **B**, **#**, **\$**, **(** i **)** i u njemu se neće pojavljivati nikakvi drugi znakovi (pa ni razmaci).

Broj znakova u izrazu će biti manji ili jednak od 100.

Zadani izraz će uvijek biti takav da se na kalkulatoru može dobiti od A i B.

Izlazni podaci

U izlaznu datoteku je potrebno zapisati niz naredbi kojima se dobiva zadani izraz, svaku u svoj red.

Napomena: rješenje ne mora biti jedinstveno.

KALK

Test primjeri

KALK.IN

(B\$A)

KALK.OUT

SWAP
DOLLAR

KALK.IN

(A\$(A#B))

KALK.OUT

SWAP
DUP
DUP
ROT
HASH
DOLLAR
SWAP
DROP

KALK.IN

((A\$B)#(B\$A))

KALK.OUT

SWAP
DUP
DUP
ROT
ROT
DROP
DUP
DUP
ROT
ROT
ROT
DROP
SWAP
ROT
ROT
DOLLAR
DUP
ROT
ROT
ROT
DROP
SWAP
DOLLAR
HASH

Pojašnjenje (ne treba ispisivati)

Test primjer #2 (ostali test primjeri slično)

	2: A	SWAP	DUP	DUP	ROT	HASH	DOLLAR	SWAP	DROP
1: B	2: B	3: B	4: B	4: A	3: A		2: A	2: (A\$(A#B))	1: (A\$(A#B))
	1: A	2: A	3: A	3: A	2: A	2: A	1: (A\$(A#B))	1: A	
		1: A	2: A	2: A	1: (A#B)				
			1: A	1: B					