

Descrierea generală a problemelor

Nr. crt.	Denumirea problemei	Restricția referitoare la volumul utilizat de memorie	Restricția referitoare la timpul de execuție, secunde	Punctajul alocat problemei
1.	Calibrarea senzorilor	$\leq 1\text{Mb}$	$\leq 0,1$	100
2.	Noua investigație a lui Sherlock	$\leq 1\text{Mb}$	$\leq 0,1$	100
3.	Eficiența salarizării în proiecte IT	$\leq 4\text{Mb}$	$\leq 0,2$	100

Notă. În caz de egalitate de punctaj, prioritate se va da concurentului care a obținut punctajul respectiv primul.

Calibrarea senzorilor

Cercetătorii de la laboratorul de cercetări nucleare au decis să calibreze senzorii acceleratorului de particule. Pentru aceasta, ei fotografiază particulele care trec prin camera acceleratorului. În imagini sunt vizibile urmele particulelor, care sunt descrise de funcția $f(x) = a \cdot x^2 + b \cdot x + c$, unde x - coordonata particulei, a , b , c - coeficienții funcției. În imagine oamenii de știință analizează urma particulei care trece printr-un anumit interval de coordonate $[M, N]$. Pentru calibrarea senzorilor, oamenii de știință doresc să găsească minimumul funcției $f(x)$ pe acest interval și să compare această valoare cu datele senzorilor. Ajutați-i să facă acest lucru.

Sarcină. Elaborați un program, care va determina minimumul funcției $f(x)$ pe intervalul $[M, N]$.

Date de intrare. Prima linie a intrării standard va conține două numere întregi M și N - limitele intervalului $[M, N]$. În a doua linie sunt scrise trei numere întregi a , b , c - coeficienții funcției $f(x)$.

Date de ieșire. În prima linie a ieșirii standard afișați coordonatele punctului minim al funcției $f(x)$ pe intervalul $[M, N]$ cu o precizie de 10^{-6} .

Restricții. $-1000 \leq M, N \leq 1000$; $-1000 \leq a, b, c \leq 1000$ Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `calibrare.pas`, `calibrare.c` sau `calibrare.cpp`.

Exemplu

Intrare

```
-1 2
1 0 0
```

Ieșire

```
0.000000 0.000000
```

Intrare

```
-1 2
1 0 -1
```

Ieșire

```
0.000000 -1.000000
```

Intrare

```
-1 2
1 -2 1
```

Ieșire

```
1.000000 0.000000
```

Noua investigație a lui Sherlock

Într-un oraș aglomerat a fost raportat furtul unui dispozitiv inovator. Detectivul Sherlock, cunoscut pentru rezolvarea celor mai complicate cazuri, a fost desemnat să investigheze cazul. Dispozitivul furat este dotat cu un tracker care, odată activat, emite semnale ce indică direcția deplasării hoțului: Nord (N), Est (E), Sud (S) sau Vest (V). Trackerul a fost activat de la distanță în momentul depistării furtului. În urmărirea sa, detectivul Sherlock a reușit să adune o serie de m semnale din ultimele minute, formând o secvență de indicii cum ar fi "VS?EE??", unde semnele de întrebare "?" indică semnale corupte pentru care direcția nu a putut fi determinată precis. Detectivul cunoaște foarte bine orașul și știe că infractorul, în încercarea de a evita capturarea rapidă, alege să se deplaseze exclusiv prin spațiile publice ale orașului pentru a se amesteca cu mulțimea. Orașul poate fi prezentat în forma unei grile $k \times l$, unde simbolul "#" (diez) semnalează zonele inaccesibile (clădiri, ziduri etc.), iar "." (punct) spațiile publice deschise pe care infractorul le poate folosi pentru deplasare. Un exemplu al reliefului orașului este:

```

...##...
..#.#.#..#
..#.....##
.##...#..
.....#....

```

Fără a cunoaște locația exactă unde se afla dispozitivul furat la conectarea trackerului, dar având la dispoziție harta orașului și indiciile colectate, detectivul trebuie să identifice toate locațiile posibile în care se află hoțul cu dispozitivul furat în momentul actual, pentru a organiza o captură eficientă și a recupera dispozitivul furat. Un semnal al trackerului corespunde unei mișcări către celula vecină (sus - Nord, jos - Sud, stânga - Vest, dreapta - Est) în funcție de direcție. Hoțul nu se poate mișca în zonele inaccesibile.

Sarcină. Elaborați un program, care va determina numărul de locații actuale posibile cunoscând harta orașului și indiciile colectate.

Date de intrare. Intrarea standard conține pe prima linie trei numere naturale k , l , m , separate prin spațiu. Pe următoarele k linii sunt l caractere „#” și „.” reprezentând harta orașului. Pe ultima linie a intrării sunt descrise semnalele interceptate de Sherlock - un șir de m caractere, toate aparținând mulțimii {N, E, S, V, ?}.

Date de ieșire. Ieșirea standard va conține o linie cu un număr întreg ce reprezintă numărul de locații curente distincte în care se poate afla hoțul.

Restricții. $1 \leq k, l \leq 500$; $1 \leq m \leq 5000$. Setul de date inițiale admite o singură soluție a problemei. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `investigatie.pas`, `investigatie.c` sau `investigatie.cpp`.

Exemplu

Intrare

```
3 3 3
..#
#..
...
VEE
```

Ieșire

```
1
```

Intrare

```
4 4 3
..#.
#..#
....
.##.
SES
```

Ieșire

```
2
```

Explicație.

În primul exemplu, o mișcare la Vest, după care să fie posibile două mișcări la Est, sunt posibile doar dacă hoțul se afla în poziția (2; 1) în momentul conectării trakerului. După mișcărilor VEE, hoțul va fi în poziția (2; 2). Respectiv există o singură poziție posibilă a hoțului.

În al doilea exemplu, ținând cont de faptul că trakerul a interceptat mișcărilor SES (sud – est – sud) avem posibile două situații:

- a) Prima posibilitate reflectă situația când interceptarea a început în poziția (0; 1), apoi hoțul a mers spre sud în poziția (1; 1), apoi spre est în poziția (1; 2), după care s-a dus iar spre sud în poziția (2;2). Deci poziția posibilă a hoțului ar putea fi (2; 2).

	0	1	2	3
0	.	.	#	.
1	#	.	.	#
2
3	.	#	.	#

- b) A doua posibilitate reflectă situația când interceptarea a început în poziția (1; 1), apoi spre sud în poziția (2; 1), apoi spre est în poziția (2; 2), după care s-a dus iar spre sud în poziția (3;2). Deci poziția posibilă a hoțului ar putea fi (3; 2). Deci răspunsul este 2 – două poziții posibile ale hoțului.

	0	1	2	3
0	.	.	#	.
1	#	.	.	#
2
3	.	#	.	#

Eficiența salarizării în proiecte IT

Mihai este administrator al unei companii IT. Compania sa se confruntă cu provocarea de a finaliza n proiecte majore. Pentru a finaliza un proiect i este necesar de un volum de muncă estimat la A_i ore.

Compania poate angaja m specialiști IT, fiecare specialist j are angajament contractual de a lucra un număr maxim de B_j ore. Fiecare specialist dorește să primească salariul integral, chiar dacă numărul efectiv de ore lucrate este mai mic decât B_j . Un specialist poate contribui la realizarea mai multor proiecte, alocând un număr diferit de ore fiecărui proiect, dar un proiect este considerat finalizat doar dacă la dezvoltarea sa contribuie cel puțin k specialiști și numărul total de ore de muncă alocate sunt, în sumă, egale cu A_i . Contribuția fiecărui specialist la un proiect este exprimată printr-un număr întreg de ore.

Scopul lui Mihai este să angajeze o echipă de specialiști (nu neapărat toți m) astfel încât să asigure finalizarea tuturor proiectelor și să reducă la minim totalul orelor pentru care se plătește fără ca acestea să fie efectiv lucrate. Acest lucru necesită o strategie optimă pentru distribuirea resurselor umane pe proiecte, cu scopul de a maximiza eficiența operațională și de a minimiza costurile nejustificate.

Sarcină. Elaborați un program, care va determina numărul minim de ore achitate în plus echipei de specialiști IT pentru a asigura finalizarea cu succes a tuturor proiectelor, minimizând numărul de ore achitate nejustificat.

Date de intrare. Prima linie a intrării standard conține trei numere naturale n, m, k , separate de un spațiu. A doua linie conține n numere întregi A_i , separate de un spațiu. Pe a treia linie sunt m numere întregi B_j , separate de un spațiu.

Date de ieșire. Ieșirea standard va conține o linie cu numărul total de ore pe care specialiștii IT nu le lucrează efectiv, dar sunt achitați de companie. Dacă, folosind regulile de mai sus, nu există nici o posibilitate de a finaliza cele n proiecte propuse se va afișa „Imposibil”.

Restricții. $1 \leq m, n, k, A_i, B_j \leq 300$. Setul de date inițiale admite o singură soluție a problemei. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `salarizare.pas`, `salarizare.c` sau `salarizare.cpp`.

Exemplu

Intrare

```
1 2 2
6
4 5
```

Ieșire

```
3
```

Intrare

```
1 1 3
4
4
```

Ieșire

```
Imposibil
```

Explicație.

În primul exemplu este necesar de a finaliza un proiect și pot fi angajați cel mult doi specialiști. Pentru a finaliza proiectul sunt necesari doi specialiști IT. Deoarece pentru finalizarea proiectului sunt necesare 6 ore, iar specialiștii sunt contractați pentru $4 + 5$ ore, vom obține că 3 ore specialiștii nu le vor lucra ($9 - 6$), dar salariul îl vor primi.

În al doilea exemplu este necesar de a finaliza un proiect și poate fi angajat un specialist. Pentru a finaliza proiectul sunt necesari 3 specialiști, dar avem doar un singur specialist, deci va fi imposibil să finalizăm proiectul.