

Ziua 1**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.	Arena	≤ 256 Kb	$\leq 0,002$	100
2.	Bare	≤ 4256 Kb	$\leq 2,700$	100
3.	Sobolul	≤ 37 Mb	$\leq 0,790$	100

Notă. În caz de egalitate de punctaje, mai bune vor fi considerate soluțiile cu un timp de execuție mai mic. În caz de egalitate și a timpilor de execuție, mai bune vor fi considerate soluțiile ce utilizează un volum mai mic de memorie.

Arena

Primii pământeni care au ajuns pe Marte au găsit pe această planetă mai multe ruini străvechi. În particular, ei au găsit și ruinele unor arene pe care cândva, cel mai probabil, luptau gladiatorii marțieni. Fiecare din aceste arene are forma unui poligon regulat¹, dimensiunile și numărul de laturi variind de la o arenă la alta. Fiecare colț al arenei (vârful poligonului) era indicat printr-un stâlp. Între stâlpi se întindeau frânghiile (laturile poligonului) ce mărgineau arena propriu-zisă.

Recent arheologii spațiali au mai descoperit pe Marte încă o astfel de arenă. Din nefericire însă, din toți stâlpii ce indicau colțurile arenei s-au păstrat doar trei, restul fiind distruși de furtunile de pe Marte.

În prezent, arheologii spațiali încearcă să determine numărul minim de stâlpi n pe care le-ar fi putut avea arena recent descoperită.

Sarcină. Elaborați un program, care, cunoscând coordonatele carteziene (x, y) ale fiecăruia din cei trei stâlpi, determină numărul minim de colțuri n pe care le-ar fi putut avea arena.

Date de intrare. Intrarea standard conține trei linii. Pe fiecare din aceste linii se află câte două numere reale separate prin spațiu. Linia i a intrării standard conține coordonatele carteziene (x_i, y_i) ale stâlpului i .

Date de ieșire. Ieșirea standard va conține pe o singură linie numărul întreg n .

Restricții. $-1000 \leq x_i, y_i \leq 1000$; $n \leq 100$. După punctul zecimal, numerele reale x_i, y_i pot avea cel mult șase cifre zecimale. 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 `arena.pas`, `arena.c` sau `arena.cpp`.

Exemplu.

Intrare

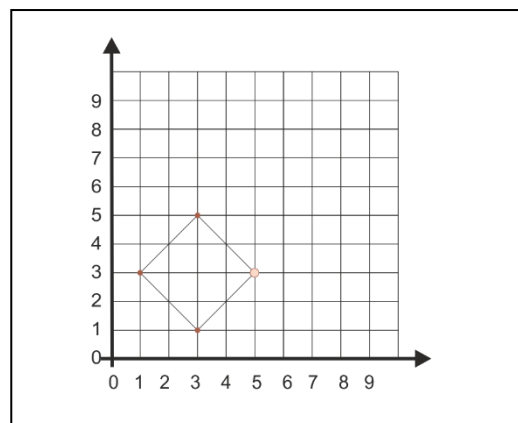
```
1.000000 3.000000
3.000000 1.000000
3.000000 5.000000
```

Ieșire

```
4
```

Explicație.

Laturile unui poligon regulat au lungimi egale. Deoarece distanțele între cei trei stâlpi rămași sunt diferite, este logic să presupunem existența a încă cel puțin a unui stâlp (vârf), care era folosit pentru a delimita laturile arenei. Astfel, adăugarea vârfului în punctul cu coordonatele $(5, 3)$ duce la formarea unui pătrat, care are toate laturile de lungime egală și toate unghiurile congruente (vezi figura alăturată). În general, oricare poligon regulat ce va conține în calitate de vârfuri cei trei stâlpi rămași, va avea $4k$ laturi, unde $k = 1, 2, 3, \dots$.



¹ *Poligon regulat* – poligon convex, care are toate laturile congruente și toate unghiurile congruente.

Bare

Se consideră un semifabricat în formă de bară, cu lungimea L . Din acest semifabricat trebuie tăiate bare de lungimi mai mici, de dimensiuni prestabilite, selectate din mulțimea de dimensiuni $\{D_1, D_2, \dots, D_k\}$.

Evident, există mai multe variante posibile de tăiere, ele putând fi formate atât prin alegerea din mulțimea $\{D_1, D_2, \dots, D_k\}$ a anumitor dimensiuni, cât și prin stabilirea numărului de bare n_1, n_2, \dots, n_k ce vor fi tăiate în conformitate cu fiecare din dimensiunile alese.

Pentru a utiliza eficient materia primă și a proteja mediul ambiant, tehnologul dorește să aleagă astfel de dimensiuni D_i și cantități de bare de tăiat n_i , încât lungimea R a restului de semifabricat, rămas după tăierea barelor, să fie minimă.

De exemplu, pentru $L = 10, k = 2, D_1 = 2, D_2 = 4$ se va obține valoarea minimală $R = 0$ dacă se va utiliza una din următoarele modalități de tăiere:

- 1) vor fi tăiate o bară de dimensiunea $D_1 = 2$ și două bare de dimensiunea $D_2 = 4$;
- 2) vor fi tăiate cinci bare de dimensiunea $D_1 = 2$;
- 3) vor fi tăiate trei bare de dimensiunea $D_1 = 2$ și o bară de dimensiunea $D_2 = 4$.

Sarcină. Elaborați un program, care, cunoscând lungimea semifabricatului L și dimensiunile prestabilite $\{D_1, D_2, \dots, D_k\}$ ale barelor ce ar putea fi tăiate, calculează lungimea minimal posibilă R a restului de semifabricat.

Date de intrare. Prima linie a intrării standard conține numerele întregi L și k , separate prin spațiu. Linia a doua a intrării standard conține numerele întregi D_1, D_2, \dots, D_k , separate prin spațiu.

Date de ieșire. Ieșirea standard va conține pe o singură linie numărul întreg R .

Restricții. $0 < L \leq 4000000$; $k \leq 1000$; $0 < D_i < L, i = 1, 2, \dots, k$. Lungimea L și dimensiunile D_i sunt numere întregi. 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 `bare.pas`, `bare.c` sau `bare.cpp`.

Exemplu 1.

Intrare

```
10 3
2 3 4
```

Ieșire

```
0
```

Exemplu 2.

Intrare

```
10 2
6 7
```

Ieșire

```
3
```

Exemplu 3.

Intrare

```
39999 7
30 40 50 60 70 80 90
```

Ieșire

```
9
```

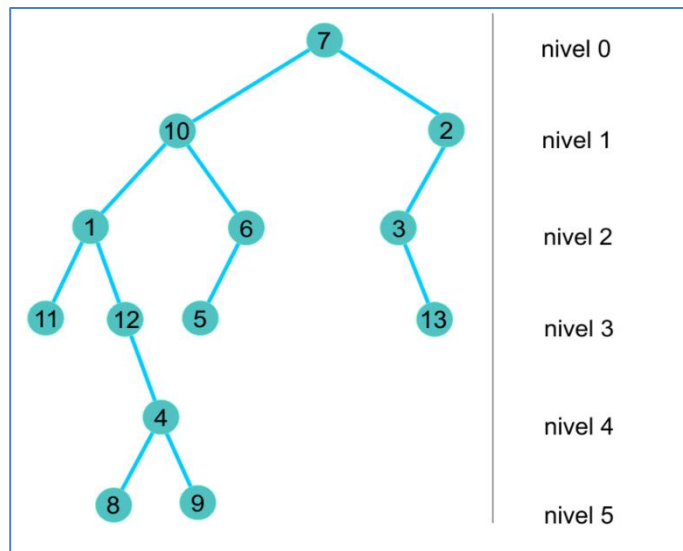
Sobolul

Ion se confruntă cu o mare problemă. În gradina lui a apărut un sobol. Deși el nu mănâncă plante, movilele făcute de el strică aspectul grădinii și Ion ar dori să îl alunge. Pentru început, el a decis să afle cum este organizată vizuina sobolului în cauză.

În urma cercetărilor efectuate cu ajutorul unui microbot autonom, Ion a constatat că vizuina are o structură arborescentă, ce include mai multe ascunzători subterane (vezi figura alăturată). Ion a numerotat ascunzătorile în cauză într-o ordine arbitrară, prin numerele consecutive 1, 2, 3, ..., N .

Sobolul se deplasează între ascunzători prin galerii rectilinii. Intrarea în vizuină este prin ascunzătoarea aflată cel mai aproape de suprafața pământului (nivelul 0).

Fiecare din ascunzătorile de pe nivelul i , $i = 1, 2, 3$ ș.a.m.d., este obligatoriu legată printr-o galerie cu o ascunzătoare de nivelul $(i-1)$ și cu cel mult două ascunzători de nivelul $(i+1)$. Galerile sunt săpate oblic, prin urmare pot exista următoarele cazuri:



- ascunzătoarea de pe nivelul i nu are galerii spre ascunzătorile de pe nivelul $(i+1)$;
- ascunzătoarea de pe nivelul i este legată cu o singură ascunzătoare de pe nivelul $(i+1)$, prin o galerie oblică spre stânga;
- ascunzătoarea de pe nivelul i este legată cu o singură ascunzătoare de pe nivelul $(i+1)$, prin o galerie oblică spre dreapta;
- ascunzătoarea de pe nivelul i este legată cu două ascunzători de pe nivelul $(i+1)$, prin două galerii oblice: una spre stânga, iar a doua – spre dreapta.

Pentru a alunga sobolul, Ion a decis să planteze în vizuina lui două dispozitive repelente – generatoare ultrasonore minusculă, undele emise de care, așa cum speră Ion, ar îndepărta astfel de animale. După lungi meditații, Ion a ajuns la concluzia că cea mai bună variantă de plantare a dispozitivelor repelente ar fi amplasarea lor în ascunzătoarea cea mai din stânga și în ascunzătoarea cea mai din dreapta ale unuia din nivelele vizuinei. Însă, întrucât numerotarea ascunzătorilor a fost făcută fără a respecta o anumită ordine, iar datele transmise de microbot au o structură specifică, Ion nu știe care ar fi numerele de identificare ale ascunzătorilor aflate la marginile nivelului selectat.

De exemplu, dacă Ion ar decide să planteze dispozitivele repelente în ascunzătorile aflate la marginile nivelului 3, numerele de identificare ale acestora ar fi 11 și 13.

Sarcină. Elaborați un program, care, în baza datelor colectate de micro robot, determină numerele de identificare ale ascunzătorilor aflate la marginile nivelului propus K .

Date de intrare. Prima linie a intrării standard conține numerele întregi N și K , separate prin spațiu. Fiecare din următoarele $N-1$ linii ale intrării standard conține câte două numere întregi și unul din caracterele S, D, separate prin spațiu. Numerele în cauză indică perechi de ascunzători ce sunt legate printr-o galerie: primul număr – pe cea care se află pe nivelul i , iar al doilea – pe cea care se află pe nivelul $(i+1)$. Caracterul S indică faptul că ascunzătoarea de pe nivelul $(i+1)$ se află în stânga,

iar caracterul D – că ascunzătoarea în cauză se află în dreapta în raport cu ascunzătoarea de pe nivelul i .

Date de ieșire. Ieșirea standard va conține pe o singură linie două numere întregi separate prin spațiu – numărul de identificare a ascunzătorii din stânga și numărul de identificare a ascunzătorii din dreapta ale nivelului K .

Restricții. $3 \leq N \leq 1000000$; $0 \leq K \leq 500$. 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 `sobol.pas`, `sobol.c` sau `sobol.cpp`.

Exemplu.*Intrare*

13	3
4	8 S
10	6 D
7	2 D
4	9 D
1	12 D
6	5 S
12	4 D
2	3 S
3	13 D
1	11 S
10	1 S
7	10 S

Ieșire

11	13
----	----