

BARAJ

Exercițiul 1

ex

Sursa : ex.c, ex.cpp sau ex.pas

În una dintre zile, la Olimpiada de Informatică se organizează **P** excursii atractive. La aceste excursii participă în total **N** persoane. Pentru simplitate, persoanele au fost numerotate de la **1** la **N**, primele **K** persoane fiind ghizii. O persoană se poate înscrie la exact una dintre cele **P** excursii organizate.

Pentru a evita surprizele neplăcute (insuficiente mijloace de transport, insuficiente locuri la restaurant, etc.) organizatorii intenționează să studieze toate configurațiile ce pot să apară în urma înscrierilor participanților, considerând totuși că în fiecare excursie va exista cel puțin un participant.

Cerință

Scrieți un program care să determine numărul de configurații distincte ce se pot obține după înscrierea celor **N** persoane la cele **P** excursii organizate, astfel încât cei **K** ghizi să fie înscriși în excursii diferite.

Date de intrare

Fișierul de intrare se numește **ex.in** și conține o singură linie pe care se află 3 numere naturale separate prin câte un spațiu: **N K P** (reprezentând numărul de persoane, numărul de ghizi și respectiv numărul de excursii).

Date de ieșire

Fișierul de ieșire **ex.out** conține o singură linie pe care se află numărul de configurații distincte.

Restricții

- $1 \leq K \leq P \leq N \leq 100$
- Într-o configurație nu contează ordinea excursiilor sau ordinea în care se înscriu persoanele la o excursie.

Exemplu

ex.in
5 3 4

ex.out
7

Explicație

Cele 7 configurații distincte sunt:

(1, 4) (2) (3) (5)

(1) (2, 4) (3) (5)

(1) (2) (3, 4) (5)

(1) (2) (3) (4, 5)

(1, 5) (2) (3) (4)

(1) (2, 5) (3) (4)

(1) (2) (3, 5) (4)

Timp maxim de execuție: 0.1 secunde/test sub Linux și 0.5 secunde sub Windows.

BARAJ

Exercițiul 2

robot

Sursa : robot.c, robot.cpp sau robot.pas

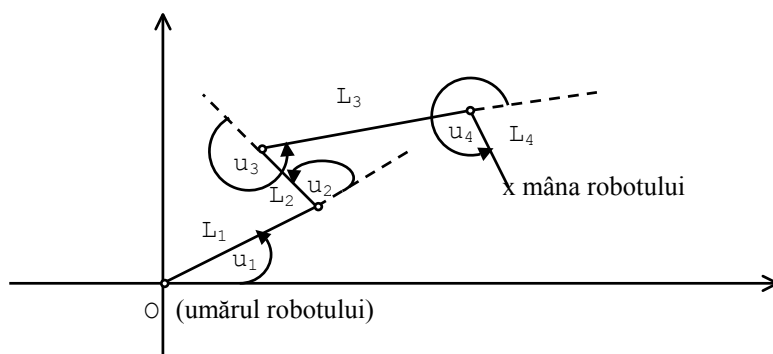
Lucrați la o firmă care produce microprocesoare. Pentru asamblarea microprocesoarelor, firma utilizează un robot constituit dintr-un singur braț. Brațul este fixat la unul dintre capete („umărul”) într-un punct plasat în centrul platformei de lucru, iar la celălalt capăt are un dispozitiv de lungime neglijabilă cu care poate „culege” componentele de pe platforma de lucru („mâna”). Brațul se poate mișca numai în plan orizontal, deasupra platformei de lucru.

Brațul este constituit dintr-o succesiune de N segmente rigide de lungimi $L_1, L_2, \dots, L_N$, conectate prin puncte de articulație. Mai exact, segmentul 1, este conectat printr-un punct de articulație în umărul robotului, segmentul 2 este conectat printr-un punct de articulație de segmentul 1, ..., segmentul N este conectat printr-un punct de articulație de segmentul $N-1$ și are la celălalt capăt „mâna”. Un punct de articulație permite rotația liberă (la orice unghi) a segmentului conectat în acel punct de articulație.

Pentru asamblarea unui microprocesor robotul trebuie să culeagă succesiv componentele acestuia de pe platforma de lucru. Fiecare componentă are o poziție bine determinată pe platforma de lucru, prin coordonatele sale relativ la un sistem de coordonate cartezian, cu centrul în umărul robotului.

Rolul dvs. în firmă este de a programa mișcările robotului. În acest scop, pentru fiecare componentă pe care robotul o va culege trebuie să specificați „configurația” brațului robotului care să permită atingerea componentei respective (mâna robotului să fie plasată deasupra poziției în care se află componenta).

Configurația brațului robotului este definită de unghiurile dintre segmentele brațului rigid.



Cerință

Scrieți un program care, pentru o poziție dată, determină o configurație pentru brațul robotului care să-i permită acestuia să culeagă componenta din poziția respectivă, dacă este posibil.

Date de intrare

Fișierul de intrare `robot.in` conține pe prima linie un număr natural N , care reprezintă numărul de segmente din care este format brațul robotului.

Pe fiecare dintre următoarele N linii se află câte un număr natural. Numărul aflat pe linia $i+1$ este lungimea celui de-al i -lea segment al brațului robotului.

Pe ultima linie se află două numere întregi x și y , separate prin câte un spațiu, reprezentând coordonatele poziției la care trebuie să ajungă „mâna” robotului.

Date de ieșire

Fișierul de ieșire `robot.out` conține o singură linie pe care se află valoarea 0 dacă nu este posibil ca mâna robotului să ajungă în poziția x, y . Dacă problema are soluție, fișierul de ieșire conține N linii. Pe linia i se află valoarea reală u_i care reprezintă unghiul dintre segmentul i și segmentul $i-1$ (pentru orice i de la 2 la N), iar valoarea u_1 reprezintă unghiul pe care segmentul 1 îl formează în umărul robotului cu axa OX .

Restricții și precizări

- $2 \leq N \leq 10000$
- $1 \leq L_i \leq 200$
- $0 \leq u_i < 360$
- $-1000000 \leq x, y \leq 1000000$
- Unghiurile se măsoară în sens trigonometric și sunt exprimate în grade.
- Programul de evaluare va verifica dacă punctul în care este plasată mâna robotului pentru configurația dată de dvs. (x_p, y_p) coincide cu punctul de coordonate (x, y). Eroarea admisă este de 0.001. Mai exact, $\max\{|x-x_p|, |y-y_p|\} < 0.001$
- Orice linie se termină cu un marcaj de sfârșit de linie (Enter).

Exemple

robot.in 3 10 5 25 15 20	Robot.out 125.6725 0 252.5424
--	---

robot.in 3 10 5 25 2 4	Robot.out 0
--	-----------------------

Timp maxim de execuție/test: 0.1 secunde pentru Linux și 0.3 secunde pentru Windows.

BARAJ

Exercițiul 3 telegraf Sursa : telegraf.c, telegraf.cpp sau telegraf.pas

Până nu demult, comunicația la distanță se făcea cu ajutorul telegrafului. Folosind telegraful, se pot transmite două tipuri de semnale: punct și linie. În general, dorim să transmitem texte formate din litere ale alfabetului latin și cifre (în total, 36 de simboluri). Trebuie deci să folosim o codificare, adică să asociem fiecăruia din cele 36 de simboluri o succesiune distinctă de linii și puncte. Pentru a putea decodifica o succesiune recepționată de linii și puncte, este necesar ca nici un simbol să nu aibă o codificare identică cu începutul codificării pentru un alt simbol. Să considerăm câteva exemple (presupunând că nu vrem să transmitem decât literele A,B,C):

Exemplul 1	Exemplul 2	Exemplul 3
A = ..	A = .--	A = .-..
B = .-	B = .-	B = -. .
C = -	C = -	C = .- .

Exemplul 1 reprezintă o codificare corectă. Exemplul 2 reprezintă o codificare greșită, pentru că începutul codificării pentru A este identic cu codificarea pentru B (deci, o secvență de genul .-- este ambiguă, putând însemna și A și BC). Exemplul 3 este de asemenea o codificare greșită pentru că începutul codificării pentru A este identic cu codificarea pentru C (o secvență precum .- .- . este ambiguă, putând însemna fie AB, fie CC).

Se știe că într-o transmisie telegrafică, punctul durează o secundă, iar linia 2 secunde. Putem calcula astfel timpul necesar transmiterii unui text.

Folosind codificarea din exemplul 1, transmiterea textului CABCA = - .. .- - .. durează 11 secunde. Observați că lungimea transmisiei se poate calcula și astfel: $2(A) + 1(B) + 2(C) = 2(..) + 1(.-) + 2(-) = 2*2 + 1*3 + 2*2 = 11$.

Cerință

Se consideră un text, dat prin frecvența apariției fiecărui simbol (dintre cele 36 considerate). Să se găsească durata minimă necesară transmiterii acelu text, folosind o codificare aleasă corespunzător.

Date de intrare

Fișierul **telegraf.in** conține o singură linie cu 36 de numere întregi nenegative, separate prin câte un spațiu, reprezentând numărul de apariții ale fiecărui simbol în textul ce trebuie transmis.

Date de ieșire

Fișierul **telegraf.out** va conține un singur număr, și anume lungimea minimă (în secunde) necesară pentru transmiterea textului.

Restricții

- nici unul din cele 36 de simboluri nu apare de mai mult de 1 000 000 de ori în textul considerat
- există cel puțin două simboluri cu număr de apariții nenul

Exemplu

telegraf.in

2 1 2 0

telegraf.out

11

Se constată că este optim să se transmită acest text folosind codificarea din Exemplul 1, obținând o lungime minimă a transmisiei de 11 secunde.

Timp maxim de execuție: 0.1 secunde pe test pentru Linux și 0.3 secunde pentru Windows

Notă: 40% dintre teste vor conține maxim 16 simboluri cu frecvență de apariție nenulă.