

**Problema 1 FactorPoștal****100 puncte**

Un factor poștal din orașul Sulina are de parcurs zilnic un traseu liniar prin cele n adrese atribuite prin sarcina de lucru.

Adresele poștale sunt numerotate de la 1 la n și se parcurg o singură dată, în ordine crescătoare a numerelor, începând cu prima adresă și terminând cu adresa n .

Principalele activități desfășurate sunt: distribuirea de corespondență, plata mandatelor poștale, a pensiilor și a alocațiilor, precum și încasarea taxelor radio-tv și a altor bunuri și servicii (timbre, scrisori recomandate, mandate poștale, etc...). Factorul poștal știe sumele pe care trebuie să le plătească, respectiv să le încaseze și de asemenea că există cel puțin o adresă de unde va încasa bani.

Într-o zi, din cauza vremii nefavorabile, cursa regulată între Tulcea și Sulina nu poate circula și din această cauză, oficiul poștal nu poate primi banii necesari pentru pensiile și alocațiile care trebuie plătite a doua zi. Pentru a rezolva situația, dirigintele poștal îi cere factorului poștal să aleagă două adrese d_i și d_j de pe traseu, astfel încât parcurgând traseul începând cu d_i și terminând cu d_j , suma în bani de adus înapoi în firmă să aibă valoarea maximă.

Factorul poștal trebuie să treacă pe la toate adresele din traseu, chiar dacă la unele trebuie să efectueze plăți. În plus, acesta este îndrumat să aleagă un traseu care acoperă cât mai multe adrese.

Cerință

Scrieți un program care determină suma maximă de încasat pe traseu, vizitând un număr maxim de adrese consecutive.

Date de intrare

Fișierul **factorpostal.IN** conține pe prima linie un număr natural n , reprezentând numărul de adrese care trebuie vizitate de factorul poștal. Pe următoarea linie se află n numere întregi, despărțite prin câte un spațiu, reprezentând sumele pe care factorul poștal ar trebui să le plătească sau să le încaseze în/din cele n adrese. Numerele pozitive reprezintă sumele care se vor încasa, iar numerele negative reprezintă sumele care se vor plăti.

Date de ieșire

Fișierul **factorpostal.OUT** va conține pe prima linie un număr natural nenul reprezentând suma maximă cu care factorul poștal se va întoarce la oficiul poștal.

Restricții și precizări

$1 \leq n \leq 1000000000$
 $-1000000 \leq \text{suma}_i \leq 1000000$ ($i=1,2,3,\dots,n$)
 $1 \leq \text{sumamax} \leq 1000000000$

Exemple

factorpostal.IN	factorpostal.OUT	Explicație
10 1 2 -5 3 4 5 -4 10 -2 9	25	Suma maximă se calculează între pozițiile 4 și 10 ($25=3+4+5-4+10-2+9$)
10 1 2 -5 3 4 5 -4 10 -2 -8	18	Suma maximă se calculează între pozițiile 4 și 8 ($18=3+4+5-4+10-2-8$)

Timp maxim de executare/test: 0.2 sec

Memorie totală: 2MB din care 2MB stiva

Dimensiunea maximă a sursei: 2KB

**Problema 2 - MERI****100 puncte**

Un fermier are un teren dreptunghiular de lungime n metri și lățime m metri, împărțit în $n \times m$ pătrate de latură 1 metru. În fiecare pătrat are plantat câte un măr. Acum este perioada culesului. Pentru că ar vrea să înlocuiască câte o zonă dreptunghiulară de meri cu cireși, a stabilit k zone dreptunghiulare pentru care vrea să determine ce cantitate de mere strânge.

Cerință

Scrie un program care să determine cantitatea minimă de mere strânsă dintr-una din zonele dreptunghiulare.

Date de intrare

În fișierul de intrare `meri.in` se găsesc pe prima linie trei numere naturale n m k cu semnificația din enunț, pe următoarele n linii se află câte m numere naturale care reprezintă numărul de kg strânse din fiecare pom, iar începând cu linia $n+2$ sunt scrise încă k linii pe care se găsesc 4 numere naturale $i1$ $j1$ $i2$ $j2$ care reprezintă coordonatele colțului stânga sus și coordonatele colțului dreapta jos pentru fiecare din cele k zone dreptunghiulare

Date de ieșire

Fișierul de ieșire `meri.out` va conține pe prima linie un număr natural ce reprezintă cantitatea minimă de mere strânse dintr-una din cele k zone dreptunghiulare.

Restricții și precizări

- $1 \leq n, m, k \leq 100$
- cantitatea de mere strânsă din fiecare pom nu depășește 100 kg.
- $1 \leq i1 \leq i2 \leq 100$
- $1 \leq j1 \leq j2 \leq 100$

Exemplu

<code>meri.in</code>	<code>meri.out</code>
4 6 3	38
3 1 10 7 5 2	
1 5 3 8 6 3	
10 1 3 8 10 8	
6 1 8 3 9 1	
2 2 3 5	
1 2 4 6	
2 1 4 3	

Timp maxim de execuție: 0.02 secunde/test.**Memorie totală: 2MB din care 2MB stiva****Dimensiunea maximă a sursei: 5 KB.**