

Problema Expresie

Fișier de intrare `expresie.in`
Fișier de ieșire `expresie.out`

Se consideră un număr natural format din n cifre. Inserând între cifrele numărului dat p operatori $+$ și q operatori $-$ se obține o expresie aritmetică. Un operator poate fi inserat doar între două cifre, deci înaintea primei cifre a numărului nu se poate plasa un operator.

Cerință

Scrieți un program care, pentru un număr dat, determină valoarea maximă a unei expresii aritmetice care se poate obține inserând p operatori $+$ și q operatori $-$ între cifrele numărului dat.

Date de intrare

Fișierul de intrare `expresie.in` conține pe prima linie numerele naturale n p q separate prin câte un spațiu, cu semnificația din enunț. Pe cea de-a doua linie se află un număr format din n cifre.

Date de ieșire

Fișierul de ieșire `expresie.out` va conține o singură linie, pe care va fi scrisă valoarea maximă a unei expresii aritmetice care se poate obține prin inserarea a p operatori $+$ și a q operatori $-$ între cifrele numărului dat.

Restricții

- $2 \leq n \leq 1\,000$
- $0 < p + q < n$
- Numărul citit nu începe cu 0.

#	Punctaj	Restricții
1	6	$n \leq 18$ și $p + q = 1$
2	9	$n > 18$ și $p + q = 1$
3	25	Rezultatul are cel mult 18 cifre și $p + q > 1$
4	60	Nu există restricții suplimentare

Exemple

<code>expresie.in</code>	<code>expresie.out</code>	Explicații
5 1 2 54321	54	$54 + 3 - 2 - 1 = 54$

Problema Pictura

Fișier de intrare `pictura.in`
Fișier de ieșire `pictura.out`

Robotul Vasile se visează artist plastic. El creează picturi 3D, pe un suport pe care îl scoate la imprimantă. O pictură 3D are formă dreptunghiulară și poate fi împărțită în „pixeli 3D”, aranjați pe n linii și m coloane, fiecare pixel 3D având o anumită înălțime. Inițial pictura este albă.

Un pixel 3D este considerat „vârf” dacă el are înălțimea strict mai mare decât înălțimile tuturor vecinilor săi. Doi pixeli 3D se consideră *vecini* dacă ei se află pe aceeași linie, pe coloane consecutive, sau se află pe aceeași coloană, pe linii consecutive.

Robotul Vasile pictează algoritmic, în modul următor: parcurge pixelii în ordinea liniilor, iar pe fiecare linie în ordinea coloanelor și dacă pixelul curent este un vârf „toarnă” peste el o nouă culoare (o culoare diferită de alb și de toate culorile utilizate până la momentul respectiv). Vom denumi culori pure, culorile pe care robotul Vasile le toarnă peste pixelii vârf. Culoarea pură se va „scurge” și va vopsi toți pixelii 3D vecini care au înălțime strict mai mică decât a vârfului vopsit, apoi se va scurge în continuare la vecinii vecinilor cu înălțimi strict mai mici ș.a.m.d., până când scurgerea nu mai este posibilă. Atunci când culoarea pură ajunge peste un pixel alb, acesta se va colora în culoarea respectivă. Dacă însă culoarea pură ajunge peste un pixel care a fost colorat anterior, culorile se vor combina și se va crea o nouă culoare. Culorile care se formează prin combinare vor fi diferite de toate culorile pure pe care robotul Vasile le toarnă peste pixelii vârf.

Deși se visează artist, robotul Vasile nu are simț estetic, ca urmare a stabilit trei criterii de evaluare a calității artistice a unei picturi: numărul maxim de culori pure care se combină pe un pixel 3D, numărul de culori distincte care apar în pictură (pure sau obținute prin combinare), respectiv dimensiunea maximă a unei zone colorate în aceeași culoare, diferită de alb. O zonă este formată din pixeli 3D cu proprietatea că, pentru oricare doi pixeli p_1 și p_2 din zona respectivă există o succesiune de pixeli 3D care începe cu p_1 , se termină cu p_2 și oricare doi pixeli consecutivi sunt *vecini*. Dimensiunea unei zone este egală cu numărul de pixeli 3D din zona respectivă.

Cerință

Scrieți un program care, cunoscând n și m (dimensiunile picturii), respectiv înălțimile pixelilor 3D, rezolvă următoarele trei cerințe:

1. determină numărul maxim de culori pure care se combină pe un pixel 3D;
2. determină numărul de culori distincte care apar în pictura creată conform algoritmului aplicat de robotul Vasile;
3. determină dimensiunea maximă a unei zone formată din pixeli 3D de aceeași culoare, diferită de alb.

Date de intrare

Fișierul de intrare `pictura.in` conține pe prima linie trei numere naturale C n m , reprezentând cerința care trebuie să fie rezolvată (1, 2 sau 3), numărul de linii, respectiv numărul de coloane ale picturii. Pe fiecare dintre următoarele n linii se află câte m numere naturale reprezentând înălțimile pixelilor 3D care constituie suprafața pe care pictează robotul Vasile (respectând ordinea liniilor, respectiv a coloanelor). Valorile scrise pe aceeași linie sunt separate prin câte un spațiu.

Date de ieșire

Fișierul de ieșire `pictura.out` va conține o singură linie pe care va fi scris răspunsul la cerința C din fișierul de intrare.

Restricții

- $3 \leq n, m \leq 300$
- Înălțimile pixelilor 3D sunt numere naturale $\leq 30\,000$.
- Pentru toate datele de test se garantează că numărul de culori pure care se combină pe un același pixel 3D este < 200 .

#	Punctaj	Restricții
1	30	$C = 1$
2	50	$C = 2$
3	20	$C = 3$

Exemple

pictura.in	pictura.out	pictura.in	pictura.out	pictura.in	pictura.out
1 5 3 8 7 8 8 20 18 11 5 30 19 4 50 2 3 40	3	2 5 3 8 7 8 8 20 18 11 5 30 19 4 50 2 3 40	7	3 5 3 8 7 8 8 20 18 11 5 30 19 4 50 2 3 40	4

Explicații

Înălțimile pixelilor 3D sunt ilustrate în figura 1, vârfurile fiind marcate îngroșat (înălțimile acestora fiind 20, 19, 50). Toți pixelii au inițial culoarea albă, marcată cu 0.

Vom colora succesiv cele 3 vârfuri. Colorând vârful cu înălțimea 20 în culoarea 1 obținem imaginea din figura 2.

Colorând vârful cu înălțimea 19 în culoarea 2 obținem imaginea din figura 3, unde am notat cu (1, 2) combinația culorilor 1 și 2.

Colorând vârful cu înălțimea 50 în culoarea 3 obținem imaginea din figura 3, unde am notat cu (1, 3) combinația culorilor 1 și 3, iar cu (1, 2, 3) combinația culorilor 1, 2 și 3.

	1	2	3
1	8	7	8
2	8	20	18
3	11	5	30
4	19	4	50
5	2	3	40

Figura 1

	1	2	3
1	0	1	1
2	1	1	1
3	0	1	0
4	0	1	0
5	1	1	0

Figura 2

	1	2	3
1	0	1	1
2	(1, 2)	1	1
3	2	(1, 2)	0
4	2	(1, 2)	0
5	(1, 2)	(1, 2)	0

Figura 3

	1	2	3
1	0	(1, 3)	(1, 3)
2	(1, 2)	1	(1, 3)
3	2	(1, 2, 3)	3
4	2	(1, 2, 3)	3
5	(1, 2, 3)	(1, 2, 3)	3

Figura 4

Sunt vizibile 7 culori distincte: 0, 1, 2, 3, (1, 2), (1, 3) și (1, 2, 3). Numărul maxim de culori pure care se combină pe un același pixel 3D este 3. Dimensiunea maximă a unei zone formată din pixeli de aceeași culoare este 4 - zona colorată în culoarea (1, 2, 3).

Problema Secv9

Fișier de intrare `secv9.in`
Fișier de ieșire `secv9.out`

Cifra de control a unui număr natural x se obține astfel:

- dacă numărul x are o singură cifră, atunci cifra de control a lui x este egală cu x ;
- dacă numărul x are cel puțin două cifre, atunci se calculează suma cifrelor lui x (să o notăm cu s); cifra de control a lui x va fi egală cu cifra de control a lui s .

De exemplu, cifra de control a numărului 175 este egală cu cifra de control a numărului 13 ($1 + 7 + 5$), care este egală cu 4 ($1 + 3$).

Fie $x_1, x_2, \dots, x_N$ un șir de N numere naturale. Două poziții i și j , cu $1 \leq i \leq j \leq N$, definesc secvența $[i, j]$ care va conține numerele $x_i, x_{i+1}, \dots, x_j$.

O secvență $[i, j]$ cu proprietatea că suma tuturor elementelor din secvență are cifra de control egală cu 9 o vom denumi `secv9`.

Cerință

Scrieți un program care, cunoscând N , numărul de elemente din șir, respectiv $x_1, x_2, \dots, x_N$, elementele din șir, rezolvă următoarele două cerințe:

1. afișează lungimea maximă a unei secvențe `secv9`;
2. afișează numărul de secvențe `secv9` din șir.

Date de intrare

Fișierul de intrare `secv9.in` conține pe prima linie două numere naturale C și N , reprezentând cerința care trebuie rezolvată (1 sau 2), respectiv lungimea șirului. Următoarea linie conține N numere naturale $x_1, x_2, \dots, x_N$, separate prin câte un spațiu, reprezentând elementele din șir.

Date de ieșire

Fișierul de ieșire `secv9.out` va conține pe prima linie un singur număr natural, reprezentând răspunsul la cerința C din fișierul de intrare.

Restricții

- $1 \leq N \leq 1\,000\,000$
- $0 \leq x_i \leq 1\,000$, pentru oricare $1 \leq i \leq N$.
- Se garantează pentru toate datele de test că există cel puțin o secvență `secv9`.

#	Punctaj	Restricții
1	8	$C = 1$ și $1 \leq N \leq 1000$
2	10	$C = 1$ și $1001 \leq N \leq 5\,000$
3	22	$C = 1$ și $5001 \leq N \leq 1\,000\,000$
4	12	$C = 2$ și $1 \leq N \leq 1000$
5	15	$C = 2$ și $1001 \leq N \leq 5\,000$
6	33	$C = 2$ și $5001 \leq N \leq 1\,000\,000$

Exemple

secv9.in	secv9.out	Explicații
1 7 1 7 6 1 11 5 9	3	Sunt două secvențe secv9 în șirul dat: • secvența [3,5], alcătuită din numerele 6, 1 și 11, are suma termenilor $18=6+1+11$, deci cifra de control este 9; • secvența [7,7], alcătuită din numărul 9, are suma termenilor 9, deci cifra de control este 9. Lungimea maximă a unei secvențe secv9 este 3.
2 7 1 7 6 1 11 5 9	2	Sunt două secvențe secv9 în șir: [3,5] și [7,7].