

Finale

Déroulement, règlement et dates importantes de la compétition sur www.infosolympiad.lu

TÂCHE 1

PÊCHE

60 POINTS

Description

Le pêcheur Bob se trouve au bord d'un fleuve divisé en n parcelles consécutives, chacune d'une longueur d'un mètre. La i -ième parcelle ($0 \leq i \leq n - 1$) contient exactement $poiss[i]$ poissons. Bob dispose de deux types de filets de pêche, g grands et p petits. Un grand filet s'étend sur d_g parcelles et un petit sur d_p . Comme il est pêcheur depuis longtemps, il sait exactement combien de poissons sont présents dans chaque parcelle, et s'il décide de mettre un filet sur un nombre de parcelles, il pêchera tous les poissons présents dans ces parcelles.



Bob veut savoir combien de poissons il peut pêcher en utilisant chaque filet au plus une fois. Or Bob n'a pas fait attention en classe d'algorithmique, donc il lui faut votre aide pour maximiser le nombre de poissons qu'il peut pêcher de cette façon.

Tâche

Vous devez implémenter la fonction `peche()`, qui reçoit toutes les informations d'en haut et retourne le nombre maximal de poissons pêchables en utilisant chaque filet au plus une fois. Elle est implémentée comme suit :

C/C++	<code>int peche(int n, int g, int p, int dg, int dp, int* poiss)</code>
Java	<code>int peche(int n, int g, int p, int dg, int dp, int[] poiss)</code>
Python	<code>def peche(n, g, p, dg, dp, poiss)</code>

Exemple d'exécution

Considérons : $n = 10$, $g = 1$, $p = 2$, $d_g = 3$, $d_p = 2$ et $poiss = [2, 0, 5, 0, 1, 2, 3, 1, 0, 1]$.

Dans ce cas, Bob peut pêcher un maximum de 14 poissons.

Contraintes

$1 \leq n \leq 1000$, $0 \leq p, g \leq n$, $0 \leq d_p \leq d_g \leq n$

Répartition des points

Sous-tâche	Points	Contraintes supplémentaires
1	11	Tout est petit : $1 \leq n \leq 10$
2	17	Il n'y a que des petits filets : $g = 0$, $1 \leq d_p \leq 10$
3	32	Aucunes



Remettez le programme sous le nom **PECHE.xxx**, avec **xxx=C(PP), PY ou JAVA**.

Description

Pour satisfaire sa faim insatiable, une souris pénètre différentes grottes pour chercher du fromage. Comme la souris a beaucoup d'expérience, elle sait exactement à quoi ressemblent les grottes et où se trouve du fromage (s'il y en a). Dans le système de grottes, toujours rectangulaires, il y a quelques murs destructibles. Ces murs peuvent être détruits à l'aide de dynamite (une barre de dynamite par mur destructible). Malheureusement, le stock de barres de dynamite de la souris se termine lentement et elle t'a demandé si tu peux déterminer le nombre minimal de barres de dynamite dont elle a besoin pour atteindre tous les morceaux de fromage atteignables dans une grotte.



La souris ne peut que se déplacer horizontalement ou verticalement et pénètre la grotte toujours à l'entrée. Pour déterminer le nombre minimal de barres de dynamite, elle va te donner les informations suivantes : la hauteur h et la largeur l des grottes et un tableau $lab[h][l]$ contenant des caractères qui décrivent l'apparence de la grotte comme suit :

Caractère	Signification
#	Mur indestructible
-	Mur destructible
.	Rien
F	Fromage
E	Entrée

Tâche

Vous devez implémenter la fonction `chasse()`, qui reçoit toutes les informations d'en haut et retourne le nombre minimal de barres de dynamite. Elle est implémentée comme suit :

C/C++	<code>int chasse(int h, int l, vector<string> lab)</code>
Java	<code>int chasse(int h, int l, String[] lab)</code>
Python	<code>def chasse(h, l, lab)</code>

Exemple d'exécution

Input	Output	Explications
$h = 4, l = 6, lab[] [] =$ ##### #...F# #E...# #####	0	La souris n'a pas besoin de dynamite afin d'atteindre le morceau de fromage.
$h = 3, l = 9, lab[] [] =$ ##### #E.F.#.#F# #####	1	La souris peut atteindre un morceau de fromage sans utiliser de dynamite. Elle peut utiliser une seule barre de dynamite pour détruire le mur #.
$h = 5, l = 9, lab[] [] =$ ##### #E...-F# #####.### #.F.#-F# #####	2	Il y'a un morceau de fromage qui n'est pas atteignable.

$h = 5, l = 9, lab[][] =$ ##### #E.....F# ##-#.#-## #.F.-..F# #####	1	
--	---	--

Contraintes

$3 \leq h, l \leq 20$ un entier naturel.

Chaque grotte est encerclée par des murs indestructibles.

Chaque grotte n'a qu'une seule entrée.

Il n'y a pas de murs destructibles horizontalement ou verticalement adjacent à des murs destructibles.

Répartition des points

Sous-tâche	Points	Contraintes supplémentaires
1	12	$3 \leq h, l \leq 5$, chaque morceau de fromage est atteignable
2	12	$3 \leq h, l \leq 10$, chaque morceau de fromage est atteignable
3	16	Aucunes



Remettez le programme sous le nom CHASSE.xxx, avec xxx=C(PP), PY ou JAVA.

GOOD LUCK. SEE YOU AT ...



**31ST INTERNATIONAL
OLYMPIAD IN INFORMATICS**
4-11 AUGUST 2019
BAKU • AZERBAIJAN