Lëtzebuerger Informatiksolympiad 2020

Finale du 14 juillet 2020 au Lycée Classique de Diekirch

⚠ **Tous les programmes doivent être réalisés sous forme d'applications console (voir remarques sur le site *www.infosolympiad.lu* sous la rubrique "Les questionnaires").**

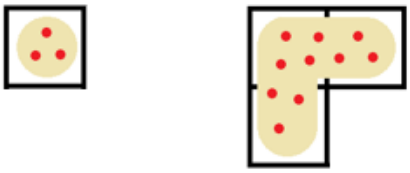
⚠ **Les formats des données ainsi que des résultats représentés dans les exemples d'exécution sont à respecter absolument.**

⚠ **Sous le fichier d'entrée on entend soit l'entrée directe des données via le clavier soit la redirection d'un fichier texte en mode console.**

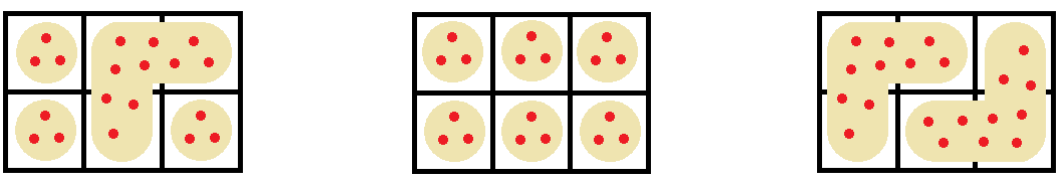
TÂCHE 1 PACKAGING 50 POINTS

Description

Lio veut acheter du gâteau chez son pâtissier préféré. Or, ce pâtissier ne produit que des gâteaux d'un seul type (du gâteau aux fraises) et qu'en deux formes, un carré de dimension ***1 x 1***, ainsi qu'un « L » composé de trois carrées de longueur ***1 x 1***, donc d'une surface de trois unités. Les deux formes de gâteaux se présentent donc comme suit :



Lio a décidé qu'elle veut rentrer avec une boîte de dimension ***n x 2*** remplie entièrement de gâteau, et elle n'a pas de préférence pour la forme du gâteau. Le pâtissier lui explique qu'il lui reste un grand nombre de gâteaux carrés, mais que malheureusement son stock de gâteaux en forme de « L » est épuisé jusqu'à ***l*** pièces. Combien de possibilités a-t-elle pour remplir la boîte ? Par exemple pour remplir une boîte de longueur 3 (***n*** = 3), il y a 11 possibilités différentes, dont les suivantes :



Si le pâtissier n'a plus qu'un seul gâteux en forme de « L » (donc ***l*** = 1), la dernière possibilité, ainsi que son image miroir ne sont plus valables, donc il ne reste plus que 9 possibilités dans ce cas.

Tâche

Écrivez un programme qui détermine le nombre de possibilités pour remplir une boîte de longueur ***n*** et de largeur ***2***, en respectant le nombre de gâteaux en forme de « L » qu'il reste. Il y a un nombre illimité de gâteaux carrés.

Contraintes

$n \in \mathbb{N}^*$ avec $1 \leq n \leq 100$

$l \in \mathbb{N}$ avec $0 \leq l \leq 67$

Entrée et sortie du programme

Entrée

Sur la première ligne, le fichier d'entrée contient un entier strictement positif, la longueur de la boîte ***n***.

Sur la deuxième ligne, le fichier contient un entier positif, le nombre de gâteaux en forme de « L » : ***l***.

Sortie

Le nombre de possibilités pour remplir une boîte ***n x 2*** avec des pièces de gâteau décrits au-dessus. Comme ce nombre peut être très grand, vous devez indiquer ce nombre modulo $10^9 + 7$.

Exemples d'exécution

Premier exemple

Fichier d'entrée	Résultat
1 67	1

Deuxième exemple

Fichier d'entrée	Résultat
2 67	5

Troisième exemple

Fichier d'entrée	Résultat
3 1	9

Quatrième exemple

Fichier d'entrée	Résultat
100 50	691082108

Répartition des points

Subtask	Points	Description
1	10	$1 \leq n \leq 6$; $l = 67$
2	10	$1 \leq n \leq 10$
3	10	$l = 67$
4	20	Pas de contraintes supplémentaires

TÂCHE 2 PÉAGE 50 POINTS

Description

Le système routier de Lioville est composé de n intersections jointes par m routes qui peuvent être traversées gratuitement. En plus de ces routes gratuites, il y a encore l routes de péage, qui peuvent être traversées seulement contre un payement (qui est le même pour chaque route). Vous êtes maintenant à l'intersection 1 et on vous attend à un restaurant près de l'intersection n. En fait, vous avez t minutes pour arriver à votre rendez-vous. Or, comme vous n'êtes pas pressé (tant que vous arrivez à temps), vous vous demandez quel est le minimum de routes de péage qu'il vous faut emprunter pour arriver à temps. Vôtre tâche est de déterminer ce minimum de routes de péage.

Entrée et sortie du programme

Entrée

La première ligne contient quatre entiers, le nombre d'intersections n, le nombre de routes gratuites m, le nombre de routes de péage l ainsi que le temps qui vous reste t.

Suivent ensuite m lignes avec trois entiers chacune. La i-ième ligne contient les deux intersections a_i et b_i jointes par la i-ième route gratuite ainsi que le temps s_i mis pour la traverser (dans n'importe quel sens).

Suivent ensuite l lignes avec trois entiers chacune. La $(i + m)$-ième ligne contient les deux intersections x_i et y_i jointes par la i-ième route de péage ainsi que le temps r_i mis pour la traverser (dans n'importe quel sens).

Sortie

Un seul entier, le nombre minimal de routes de péage qu'on doit parcourir pour arriver à temps au rendez-vous. S'il n'est pas possible d'arriver à temps, indiquez le texte « IMPOSSIBLE ».

Contraintes

Chaque intersection est connectée au système routier de la première intersection.

$1 \leq n \leq 300$ et il n'y a qu'une seule route entre deux intersections (donc soit une route gratuite, soit une route de péage).

$0 \leq l \leq 30$

$1 \leq s_i, r_i \leq 1000$

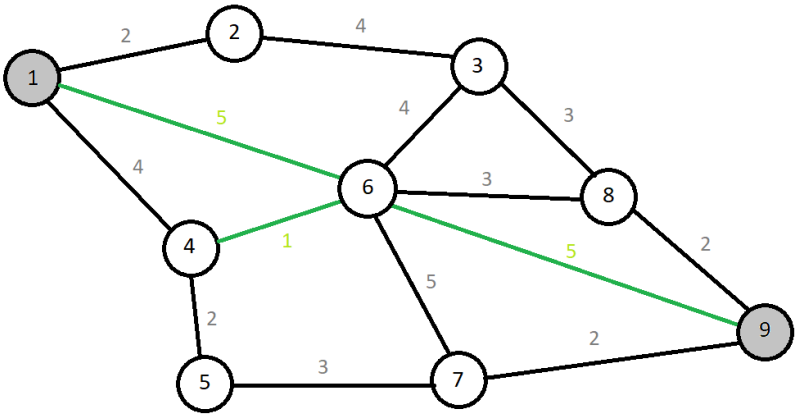
$1 \leq a_i, b_i, x_i, y_i \leq n$

$0 \leq t \leq 100\,000\,000$

Exemples d'exécution

Premier exemple

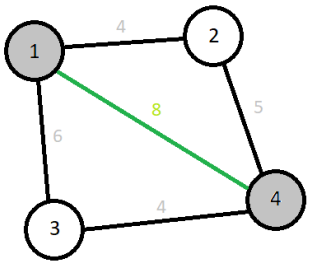
Fichier d'entrée	Résultat
9 11 3 10 1 2 2 2 3 4 3 8 3 6 8 3 1 4 4 4 5 2 5 7 3 6 7 5 7 9 2 3 6 4 8 9 2 1 6 5 4 6 1 6 9 5	1



Dans cet exemple il n'est pas possible d'arriver à temps sans utiliser les routes de péage, le minimum de temps qu'il faut sans routes de péage est de 11 minutes. Le chemin 1-6-9 passe par deux routes de péage et dure 10 minutes, or il y a moyen d'éviter la deuxième route en passant par 1-6-8-9. Donc, le résultat est 1.

Deuxième exemple

Fichier d'entrée	Résultat
4 4 1 7 1 2 4 1 3 6 2 4 5 3 4 4 1 4 8	IMPOSSIBLE



Dans cet exemple, le plus vite qu'on peut passer de l'intersection 1 à l'intersection 4, c'est en 8 minutes, en prenant la route de péage. Donc, il n'est pas possible d'arriver à temps.

Répartition des points

Subtask	Points	Description
1	5	$l = 0$
2	10	Il n'y a pas de cycles dans le réseau routier
3	10	$l \leq 5$
4	25	Pas de contraintes supplémentaires