

# Tâche 2.3: Password (password)

Auteur: Damien Galant Préparation: Damien Galant

Limite de temps: 1 s Limite mémoire: 512 MB

Puisque vous vous entraînez pour l'olympiade internationale d'informatique, votre ami informaticien accepte gentiment de vous donner accès à sa collection en ligne d'algorithmes en C++. Celle-ci est protégée par un mot de passe. Afin de vous challenger, votre ami suggère que vous trouviez vous-mêmes le mot de passe en résolvant un de ses problèmes d'algorithmique. Parviendrez-vous à relever son défi?

Son problème porte sur les chaînes de caractères formées des 26 lettres de l'alphabet, en minuscules (par exemple becp).

Deux opérations sont considérées :

- 1. Supprimer toutes les occurrences de la lettre  $l_1$  dans la chaîne de caractères
- 2. Remplacer toutes les occurrences de la lettre  $l_1$  par la lettre  $l_2$  dans la chaîne de caractères.

Ainsi, si l'opération 1 est appliquée à la chaîne aeeeaecae, avec  $l_1 = a$ , la chaîne obtenue est eeeece.

Si l'opération 2 est appliquée à la chaîne abbbabcaba avec  $l_1 = b$  et  $l_2 = e$ , la chaîne obtenue est aeeeaecaea.

Votre ami vous explique comment trouver le mot de passe de sa collection d'algorithmes :

- Une suite de caractères S vous est donnée. Il s'agit de la chaîne initiale.
- Une suite de M opérations de type 1 et 2 est donnée.
- K fois de suite, vous devez appliquer les M opérations, dans l'ordre dans lequel elles sont fournies. Si les opérations sont  $O_1, \dots, O_M$ , vous appliquez donc successivement

$$\underbrace{O_1,O_2,\cdots,O_M}_{\text{1\`ere it\'eration}},\underbrace{O_1,O_2,\cdots,O_M}_{\text{2\`eme it\'eration}},\cdots,\underbrace{O_1,O_2,\cdots,O_M}_{k\`{\text{e}me it\'eration}}.$$

À partir de S, vous appliquez  $O_1$  et obtenez une chaîne  $S_1$ , puis appliquez  $O_2$  à  $S_1$  et obtenez une chaîne  $S_2$ , et ainsi de suite.

La chaîne de caractères obtenue après ces  $K \cdot M$  opérations (c'est-à-dire  $S_{K \cdot M}$ ) est le mot de passe recherché. À vous de jouer!

#### Input

La première ligne contient les trois entiers N, M et K. La seconde ligne contient la chaîne de caractères initiale S, qui ne contient que des lettres minuscules. Les M lignes suivantes sont soit du type

- 1  $l_1$ , décrivant une opération de type 1;
- $2 l_1 l_2$ , décrivant une opération de type 2.

## Output

L'output consiste en une seule ligne contenant une chaîne de caractères formée de lettres minuscules : le mot de passe de votre ami. Il est garanti que le mot de passe à deviner est non vide.

# Limites générales

- S est formée de N lettres minuscules parmi  $\mathsf{a}\mathsf{-z}$  ;
- pour chaque opération de type 2, on a  $l_1 \neq l_2$ ;
- $-1 \leq N$ ;
- $-1 \le M$ ;
- $-N \cdot M \le 10^6$ ;
- $-1 \le K \le 10^{18}$ .

#### Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	15	K = M = 1
В	20	Il n'y a que des opérations du type 1
$\mathbf{C}$	20	$K \cdot N \cdot M \le 10^6$
D	20	Il n'y a que des opérations du type 2
E	25	Pas de contrainte supplémentaire

Remarque importante : certains nombres peuvent dépasser la capacité d'un entier de 32-bits, donc utilisez le type long long.

#### Exemple 1



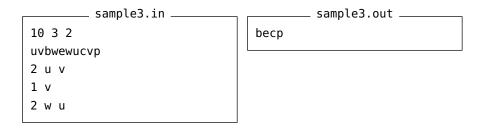
Cet exemple peut apparaître dans les sous-tâches A, B, C et E.

# Exemple 2



Cet exemple peut apparaître dans les sous-tâches A, C, D et E.

# Exemple 3



Cet exemple peut apparaître dans les sous-tâches C et E. Dans cet exemple, on doit répéter deux fois une succession de trois opérations, de types 2, 1 et 2. La chaîne initiale vaut uvbwewucvp. En appliquant les opérations, on obtient successivement vvbwewvcvp, bwewcp, bueucp, bvevcp, becp et becp. Le mot de passe est donc becp!