

# beCP

## 2022

### Tâche 1.3: Slogans (slogans)

---

Auteur: Robin Jadoul    Préparation: Bruno Ploumhans

Limite de temps: 3s    Limite mémoire: 512 MB

---

Trouver et mettre en forme un bon slogan est une tâche difficile qui demande beaucoup d'essai-erreur. Un slogan est composé de symboles : des lettres, des chiffres et des espaces. Les graphistes professionnels opèrent symbole par symbole quand ils doivent modifier la mise en forme d'un slogan, suivant les règles suivantes :

- Supprimer **le dernier symbole** a un prix fixe dépendant du dernier symbole : supprimer un espace est gratuit, supprimer une lettre minuscule coûte 5, supprimer une lettre majuscule coûte 10, et supprimer un chiffre coûte la valeur du chiffre +1.
- Ajouter un symbole **à la fin du slogan** oblige le graphiste à ajuster l'entièreté du nouveau slogan : le coût est le nouveau nombre de symboles. Par exemple, ajouter un symbole à un slogan vite coûte 1, puis en ajouter coûte 2, etc... Notez qu'ajouter un espace coûte autant qu'ajouter une lettre ou un chiffre.

La beOI vous demande de l'aide pour sa liste de slogans potentiels. Vous commencez avec une liste vide, et la beOI vous demande ensuite d'effectuer certaines des opérations suivantes.

1. Ajouter un nouveau slogan à la liste.
2. Copier un slogan existant, y ajouter des nouveaux symboles, et l'ajouter à la liste.
3. Calculer combien faire mettre à jour la mise en forme d'un slogan à un autre par un graphiste professionnel coûte.

#### Input

La première ligne contient un entier  $q$  : le nombre d'opérations à effectuer. Chacune des lignes suivantes contient un nombre  $op$ , l'opération à effectuer.

Si  $op$  est 1, le reste de la ligne contient un slogan à ajouter à la liste, entouré par ".

Si  $op$  est 2, le reste de la ligne contient  $i$  : l'indice du slogan existant, et le reste de la ligne contient des symboles à ajouter à une nouvelle copie du  $i$ -ème slogan potentiel, à ajouter à la fin de la liste.

Si  $op$  est 3, le reste de la ligne contient deux entiers  $a$  et  $b$  : les indices de deux slogans dans la liste.

Les indices  $i$ ,  $a$  et  $b$  commencent à 0 : le premier slogan a pour indice 0, le deuxième 1, et ainsi de suite... Les slogans sont garantis d'exister dans la liste.

## Output

Pour chaque opération de type 3, affichez une ligne contenant un chiffre : le coût minimum pour convertir un slogan de la  $a$ -ème séquence de symboles à un slogan pour la  $b$ -ème séquence de symboles.

## Limites générales

- $1 \leq q \leq 10^6$ , le nombre d'opérations à effectuer.
- la longueur combinée de toutes les chaînes de caractères en entrée (ce qui est entouré par ") est  $L \leq 10^6$ .

**Remarque importante :** Certaines réponses peuvent excéder la capacité d'entiers 32 bits, donc assurez-vous d'utiliser `long long`.

## Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	10	$q \leq 10^4$ , il n'y a pas d'opération de type 2, tous les slogans commencent par un symbole différent.
B	30	$q \leq 10^4$ , $L \leq 10^4$
C	60	Pas de contrainte supplémentaire

## Exemple 1

sample1.in	sample1.out
<pre>4 1 "All" 2 0 " theM pAiN" 2 0 " the medals" 3 1 2</pre>	<pre>117</pre>

Après les trois premières opérations, les slogans sont All, All theM pAiN et All the medals. Ensuite, nous devons calculer combien aller de All theM pAiN à All the medals coûte.

Le moins cher est de d'abord aller de `All theM pAiN` à `All the`, en enlevant les symboles suivants : `N`, `i`, `A`, `p`, (espace), `M` pour un coût de  $10 + 5 + 10 + 5 + 0 + 10 = 40$ .

Ensuite, aller de `All the` à `All the medals` coûte 8 pour l'espace, 9 pour le `m`, et ainsi de suite... pour un coût de  $8 + 9 + 10 + 11 + 12 + 13 + 14 = 77$ .

Le coût total est  $40 + 77 = 117$ .

*Cet exemple n'est valide que pour les sous-tâches B et C.*

## Exemple 2

sample2.in	sample2.out
<pre>4 1 "just do it" 1 "2022 Life" 3 0 1 3 1 0</pre>	<pre>85 90</pre>

Pour aller de `just do it` à `2022 Life`, le coût total est de 85.

Pour aller de `2022 Life` à `just do it`, le coût total est de 90. Notez que supprimer un `2` coûte 3 and que supprimer un `0` coûte 1, comme expliqué plus haut.

*Cet exemple est valide pour toutes les sous-tâches.*