

Programmation - Projet Allocation dynamique

30 octobre 2017

1 Consignes

1. Une fois votre projet achevé, vous devez rendre votre fichier C sur “syntax error”.
2. La date de fin du projet est celle fixée sur “syntax error”. Aucune soumission après cette date ne sera acceptée.
3. Il est formellement interdit de recopier des lignes de code d’une tierce personne. Toute tricherie avérée entraînera la note de zéro (l’entraide reste bien sûr autorisée et encouragée).

Restrictions :

1. Les seules fonctions de la libC autorisées sont `printf`, `scanf`, `malloc`, `realloc` et `free`.

2 Les substitutions

Une règle de substitution s’écrit :

$$c \rightarrow s$$

pour c un caractère et s une chaîne de caractères. La substitution est appliquée à une troisième chaîne t comme suit : On remplace toutes les occurrences de c par s dans t . Si σ est une substitution, on écrit $\sigma(s)$ pour le résultat de l’application de la substitution σ à la chaîne s .

Exemple 2.1 Soit la substitution σ donnée par $d \rightarrow v$. Alors on a $\sigma(\text{didier}) = \text{vivier}$.

Exemple 2.2 Soit la substitution σ donnée par $a \rightarrow ab$. Alors on a $\sigma(\text{ababab}) = \text{abbabbabb}$.

Une substitution n’est pas forcément composée que d’une seule règle. Dans le cas où il y a plusieurs règles, on les applique toutes.

Exemple 2.3 Soit la substitution σ donnée par :

- $a \rightarrow ab$
- $b \rightarrow a$

Alors on a :

1. $\sigma(a) = ab$
2. $\sigma(ab) = aba$

On peut appliquer plusieurs fois de suite une substitution. Pour une substitution σ on écrit $\sigma^2(a)$ pour signifier $\sigma(\sigma(a))$. De manière générale, pour un entier n on écrit $\sigma^n(a)$ pour signifier $\sigma(\sigma^{n-1}(a))$ (avec $\sigma^1(a) = \sigma(a)$).

Exemple 2.4 Soit la substitution σ donnée par :

- $a \rightarrow ab$
- $b \rightarrow a$

Alors on a :

1. $\sigma(a) = ab$
2. $\sigma^2(a) = \sigma(ab) = aba$
3. $\sigma^3(a) = \sigma(aba) = abaab$
4. $\sigma^4(a) = \sigma(abaab) = abaababa$
5. $\sigma^5(a) = \sigma(abaababa) = abaababaabaab$
6. $\sigma^6(a) = \sigma(abaababaabaab) = abaababaabaababa$

3 Le travail

Faites un programme dans un fichier `substitutions.c` qui prend les paramètres suivants :

1. Une chaîne de caractères qui donnera des règles de substitutions.
2. Une chaîne de caractères sur laquelle appliquer la substitution.
3. Un entier qui indique le nombre de fois que la substitution doit être appliquée. On pourra supposer que cet entier sera toujours positif ou nul.

Le format de la chaîne de caractère donnant les règles de substitution sera le suivant :

1. Le caractère '[' éventuellement suivi d'un ou plusieurs espaces.
2. Les règles de substitution séparées par le caractère ','. Chaque caractère ',' sera éventuellement suivi et/ou précédé d'un ou plusieurs espaces.
3. le caractère ']' éventuellement précédé d'un ou plusieurs espaces.

Le format d'une règle de substitution sera le suivant :

1. Le caractère à remplacer. On pourra supposer qu'il s'agit d'un caractère alphanumérique.
2. Les caractères '->', éventuellement suivis et/ou précédés d'un ou plusieurs espaces.
3. la chaîne de caractère à substituer. On pourra supposer que cette chaîne est non vide et qu'elle est composée de caractères alphanumériques.

On pourra aussi supposer que les règles de substitutions seront toujours non contradictoires entre elles.

Exemple 3.1 Les chaînes de caractères suivantes sont des règles de substitutions valide :

- [a -> waza]
- [a -> an, b->a]
- [c->cool,o->oclo , l -> raoul]
- [a->1, b->2, c->3, d->4, e->5, f->6, g->7, h->8]
- [1->12, 2->21]

Votre programme doit afficher le résultat de la substitution appliquée N fois à la chaîne de caractère, où N est le troisième paramètre du programme. Ce résultat devra être suivi d'un retour à la ligne. Voici un exemple d'exécution de votre programme :

```
$> gcc -o substitutions substitutions.c
$> ./substitutions "[a->ab, b->a]" a 0 | cat -e
a$
$> ./substitutions "[a->ab, b->a]" a 6 | cat -e
abaababaabaababaababa$
$> ./substitutions "[c->cool, l->raoul]" cool 1 | cat -e
cooloraoul$
$>
```

Conseils :

1. Ne paniquez pas.
2. Faites un ensemble de fonctions pour travailler sur des chaînes de caractères.
3. Pour les fonctions qui modifient des chaînes de caractères, il est conseillé de créer à chaque fois une nouvelle chaîne plutôt que de modifier les paramètres. Par exemple, si vous voulez faire une fonction qui enlève les espaces au début et à la fin d'une chaîne, il est conseillé de créer une nouvelle chaîne en mémoire qui sera égale à la première moins les espaces de début et fin, plutôt que de modifier la chaîne existante.