

## TP1 — Pretty-printer de boîtes de textes

Un pretty-printer est un programme qui sert à redisposer des données ou des programmes sous une forme agréable et facile à lire. Par exemple, un pretty-printer spécialisé pour le langage C pourrait recevoir ce programme mal disposé :

```
#include <stdio.h>
int main(int argc, char*argv[]){printf(
"Hello world!\n");return 0;}
```

et produire le même programme sous cette forme qui est bien plus lisible :

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("Hello world!\n");
    return 0;
}
```

Généralement, on peut diviser le travail du pretty-printer en trois grandes étapes : la production d'un arbre de syntaxe à partir du source ; la recherche d'une disposition agréable ; et l'impression de l'arbre de syntaxe selon la disposition choisie. Votre travail consiste à implanter un programme capable de faire la troisième partie.

Votre programme reçoit la spécification d'un texte (programme ou donnée, indifféremment) sous la forme de boîtes imbriquées et doit l'afficher en sortie sous forme ASCII. Trois sortes de boîtes sont utilisées :

- les boîtes simples : ce sont les boîtes de base et elles ne sont constituées que d'une chaîne de caractères ;
- les boîtes horizontales : elles sont constituées de deux sous-boîtes mises l'une à côté de l'autre ;
- les boîtes verticales : elles sont constituées de deux sous-boîtes mises l'une par-dessus l'autre.

Nous dirons que les deux dernières sortes de boîtes sont des boîtes complexes, par opposition aux boîtes simples. De par leur constitution, toutes les boîtes ont une largeur et une hauteur. Ces deux dimensions sont mesurées en nombres de caractères. Voici les dimensions des trois types de boîtes :

- les boîtes simples ont une largeur égale au nombre de caractères présents dans la chaîne correspondante et une hauteur de 1 caractère ;
- les boîtes horizontales ont une largeur égale à la somme des largeurs des sous-boîtes et une hauteur égale au maximum des hauteurs des sous-boîtes ;
- les boîtes verticales ont une largeur égale au maximum des largeurs des sous-boîtes et une hauteur égale à la somme des hauteurs des sous-boîtes.

La spécification du texte est donnée en notation préfixe. Une boîte simple est écrite sous la forme d'une chaîne encadrée par des guillemets (""). Le contenu de la chaîne constitue le contenu de la boîte simple. Lorsqu'un guillemet fait partie du contenu de la boîte, il doit être protégé par une barre oblique inversée ('\'). Lorsqu'une barre oblique inversée fait partie du contenu de la boîte, elle doit être protégée par une autre barre oblique inversée. Les caractères de contrôle (i.e. ceux dont le code ASCII est inférieur à 32) sont interdits dans les chaînes. Par exemple, voici une chaîne et le contenu correspondant :

```
"hat \\12\"34/"      hat \12"34/
```

Notez que cette boîte a une largeur de 11 caractères. Une boîte horizontale est écrite sous la forme d'un 'h' suivi de la sous-boîte de gauche et de la sous-boîte de droite. Au moins un espace blanc sépare les trois items. Voici un exemple de boîte horizontale :

```
h "Hell" "o world!"
```

Une boîte verticale est écrite sous la forme d'un 'v' suivi de la sous-boîte du dessus et de la sous-boîte du dessous. Au moins un espace blanc sépare les trois items. Bien entendu, les boîtes peuvent être imbriquées de façon arbitraire. Voici un exemple de boîte verticale qui contient une sous-boîte complexe :

```
v h "nw " "ne" "sw"
```

Les espaces blancs sont les caractères reconnus comme tels par la fonction C standard 'isspace'.

Le résultat attendu de votre programme est l'affichage du rectangle de texte obtenu par la combinaison des boîtes données en entrée. Il y a quelques règles supplémentaires à connaître afin d'interpréter la spécification donnée sans ambiguïté. Lors de la création d'une boîte horizontale à partir de deux sous-boîtes de hauteurs différentes, ces deux sous-boîtes doivent être alignées par le haut. Similairement, lors de la création d'une boîte verticale à partir de deux sous-boîtes de largeurs différentes, ces deux sous-boîtes doivent être alignées par la gauche. En plus, l'espace non-couvert par la juxtaposition de deux boîtes de dimensions différentes doit être rempli par des espaces (' '). Par exemple, voici une entrée décrivant des boîtes de dimensions variées et la sortie correspondante ; la sortie est encadrée afin de bien voir les espaces qui sont ajoutés :

```
h "abcd" v "ef" "ghijk"      +-----+
                               |abcdef  |
                               |   ghijk|
                               +-----+
```

La tâche à faire faire par votre programme consiste donc à lire la spécification sur l'entrée standard et à produire le rectangle de texte correspondant *avec* l'encadré sur la sortie standard. L'encadré est constitué des caractères '+' pour les coins, '-' pour les barres horizontales et '|' pour les barres verticales. N'oubliez pas le saut de ligne final après le dernier coin.

Votre programme doit être général. C'est-à-dire qu'il ne doit pas limiter la taille de l'entrée, la taille des boîtes simples (longueur des chaînes) ni la taille du rectangle de sortie. L'utilisation de l'allocation dynamique est donc obligatoire. Assurez-vous d'éviter les pointeurs fous et les fuites de mémoire. Toute la mémoire allouée dynamiquement durant l'exécution doit avoir été libérée avant que le programme ne se termine. Vous n'avez pas à vérifier la syntaxe de la spécification : les spécifications utilisées pour la correction seront toujours syntaxiquement valides. Cependant, faire tout de même une vérification de la syntaxe pourrait vous être utile lors du développement du programme.

Vous devez remettre votre programme en un seul fichier (nommé 'tp1.c') d'ici le 11 février 15h30 à l'aide de la commande suivante :

```
remise ift2030 tp1 tp1.c
```

N'oubliez pas de mettre vos noms au début du programme. Vous devez faire le travail en équipes de deux. Notez que le style de programmation est important : votre programme doit être bien structuré et bien commenté. Vous devez aussi remettre un rapport au début de la séance de démonstration du 11 février. Donnez une description concise des étapes effectuées par votre programme, des structures de données employées et des algorithmes. Ne vous éternisez pas : deux pages de densité comparable à celles-ci devraient suffire. La pénalité pour les retards est de 10% par jour.