Rope - Part 2.

Julien Allali : allali@univ-mlv.fr

1 Introduction.

La but de cette deuxième partie est de transformer la classe Rope de façon à la rendre générique vis-à-vis des objets qu'elle contient. De plus, on va externaliser la gestion mémoire des TableauxPlus à l'aide des SmartPointers. Enfin, vous devez implémenter un petit éditeur de texte utilisant vos Ropes.

2 La classe ***Rope***< ***T***>.

Dans un premier temps, transformer votre classe Rope de façon à ce que les objets contenus soient un argument template. Puis, examiner avec attention votre code afin de déterminer quelles sont les méthodes (contraintes) que doit respecter le type pris en argument *template*. Celles-ci devront être clairement spécifiées dans le rapport.

3 Les SmartPointers

Afin d'externaliser la gestion des TableauxPlus, on va implémenter des SmartPointer. Pour avoir un descriptif des SmartPointers, vous pouvez vous référez à la section 3 du sujet du projet de l'année 2003. Brièvement, vous devez implémenter une classe SmartPointer qui contiendra 3 champs:

- `T *t` : le pointeur sur l'objet interne.
- `int *c` : le compteur de référence.
- `int d` : le décalage.

Le deuxième champs (`int *`) est ce qu'on appelle le compteur de références, plusieurs SmartPointers pointant sur le même objet doivent partager le même compteur de référence. Ce champs joue le rôle du premier indice dans les TableauxPlus. Si vous le souhaiter, vous pouvez tout à fait mettre le premier et deuxième champs dans une classe séparée. Le troisième champs indique un décalage vis-à-vis du pointeur t. En effet, pour un SmartPointer donné, on veut pouvoir se décaler comme s'il pointait sur un tableau. Par exemple:

```
char *str= new char [10];
sprintf(str,''la chaine'');
SmartPointer<char> p = str; //p.d == 0
cout<<*p; // Affiche l
p++;      //p.d==1
cout<<*p; // Affiche a
p+=5      //p.d==6
cout<<*p; // Affiche i
```

Chaque cellule de la liste chaînée d'une Rope contiendra un SmartPointer de T qui pointera sur un tableau de T ainsi que la taille du tableau associé à ce pointeur (on n'a plus besoin du décalage ainsi que de la gestion mémoire, qui sont exportés dans les SmartPointers).

Dans vos SmartPointers vous devez implémenter:

- $operator*$
- $operator- >$
- $operator+$
- $operator + +$
- $operator + +(int)$
- $operator - -(int)$
- $operator - -$
- $operator-$
- $operator =$
- Constructeur par recopie, par défaut.

Libre a vous d'ajouter d'autre fonctionnalité à vos SmartPointer comme des exceptions telles que *ArrayOutOfBoundException* ou encore *NullPointerException*.

4 Éditeur de Texte

Vous devez implémenter un éditeur de texte minimaliste avec vos Ropes. Cet éditeur devra permettre:

- L'ajout d'une ligne après/avant la ligne en cours.
- La suppression de la ligne en cours.
- Le remplacement d'un texte par un autre dans tout le texte.
- Le remplacement d'un texte par un autre dans tout la ligne en cours.

- La modification de la ligne en cours.
- L'affichage du texte.
- L'affichage de la ligne en cours.
- La sauvegarde du texte en cours dans un buffer.
- Le chargement d'un buffer comme le texte courant.
- L'ajout d'un buffer après/avant la ligne en cours.

L'implementation devra impérativement utilisée une Rope de Rope de caractère pour la gestion du texte. Voici un exemple d'utilisation de l'éditeur:

```
$./editor
*** Welcome to blabla  ***
** enter ''h'' for help **
>h
Commands:
el edit the current line
al add a line after the current line
bl add a line before the current line
lt list the text
sl set the current line
dl remove the current line
vl view the current line
sb save current text in a buffer
lb load a buffer
ab add a buffer after the current line
bb add a buffer before the current line
vb view buffers
...
>lt
>al
first text
>lt
0:first text
>vl
0:first text
>bl
second text
>lt
0:second text
1:first text
>vl
1:first text
>al
```

```
toto
>lt
0:second text
1:first text
2:toto
>sb
exemple
>sl
1
>dl
>lt
0:second text
1:toto
>sl
0
>ab
exemple
>lt
0:second text
1:second text
2:first text
3:toto
4:toto
>sb
resultat
>vb
resultat:
second text
second text
first text
toto
toto

exemple:
second text
first text
toto
>q
To save your work, give me a path (blank for no saving):sav
bye bye
$./editor sav
...
```

Vous êtes complètement libre quand au choix de l'interface.

5 A Faire

Vous devez écrire l'éditeur de texte décrit ci-dessus en utilisant vos Ropes génériques munies de SmartPointers. Les sources de votre programme devront être regroupées dans une archive qui contiendra aussi un fichier README décrivant la procédure à suivre pour compiler et utiliser votre projet.

Bien évidement, les groupes devront rester identiques aux groupes pour la première partie. L'archive décrite plus haut devra être rendue par mail (allali@univ-mlv.fr) avant le Jeudi 1 Avril 2004 à 12:00 (midi).

Un rapport de quelques pages (10 max.) devra présenter brievement votre code. Vous devrez entre autre expliquer à travers des exemples, le fonctionnement des Ropes. Pensez aussi à presenter les complexités associées aux differentes opérations. Ce rapport devra être envoyé par mail (allali@univ-mlv.fr) avant le Samedi 3 Avril 2004 à 12:00 (midi).

Enfin, des soutenances auront lieu le Mardi 6 Avril 2004. Celle-ci consisteront en 10 minutes de présentation suivies de 5 minutes de questions individuelles. La date exacte sera précisée sur le forum (http://ankh.univ-mlv.fr/˜metaone/cplusplus/).