Utilisation de la STL

Christophe Labourdette

octobre 2016

1 Un générateur de mots

On veut construire en utilisant les $string$, un programme générant des mots à partir de l'alphabet classique, en utilisant une notation classique, comme, k{(3)abc}pmi, pour générer le mot kabcabcabcpmi, un peu à la manière d'un automate fini.
l'alpahbet est constitué des lettres et des chiffres uniquement (a-z, A-Z, 0-9).
Le générateur se lit de gauche à droite, entre [], on peut donner une liste de caractères possible, mais optionnels :
exemple, t[aei], donne les mots suivants : t, ta, te, ti.
Un bloc est placé entre { } et peut être répété, n fois grâce à l'option (n) ou bien de p à q fois avec l'option (p-q). L'option se place toujours au début du bloc.
m{(0-2)hy} donne les mots, m, mhy, mhyhy.
On placera les mots générés dans un vector.
Vous écrirez ensuite une fonction résumant l'alphabet résultat : taille et nombre de mots par nombre de lettres.

2 vector et list

Ecrire un programme utilisant les classes $vector$ et $list$ et réalisant (dans cet ordre précis) les actions :
- Création d'un vecteur $v1$, dont les composantes sont les 20 premiers termes de la suite de Fibonacci
- Parcourir $v1$, pour en afficher les composantes à l'écran.
- Copier $v1$ dans une liste $l1$
- Enlever de $l1$ tous les éléments impairs
- Créer une liste $l2$ dont le contenu est la concaténation de $v1$ et $l1$
- Classer la liste $l2$ par ordre décroissant
- Retire tous les doublons de $l2$
- Appliquer à chaque élément de la liste l2 le polynome $x^2 + 4x + 1$
- Retirer et afficher le premier élément de la liste $l2$
- Retirer et afficher le dernier élément de la liste $l2$

3 Quelques noms de fichier

Ecrivez un petit programme qui prend en argument des noms de fichiers et remplace ou ajoute selon la nécessité le suffixe ".nul" aux noms. (on pourra utiliser l'une des fonction find) Par exemple l'exécution de
./gronul tp.cpp frigo nb.pdf ts.gh.rem artur bernouilli.h
doit donner :

```
tp.cpp ==> tp.nul
frigo ==> frigo.nul
nb.pdf ==> nb.nul
ts.gh.rem ==> ts.gh.nul
artur ==> artur.nul
bernouilli.h ==> bernouilli.nul
```

4 histogramme

Ecrivez une fonction histogramme, prenant en paramêtres, un vecteur arbitrairement grand contenant des valeurs, un entier représentant le nombre de barre, les bornes a et b d'un intervalle. la fonction histogramme retournera un vecteur de taille n comprenant le nombre de valeurs dans chaque sous-intervalle de $[a, b]$.
On testera le programme avec des valeurs générées aléatoirement, selon différentes lois.
On pourra également avoir différentes version de la fonction selon le type des valeurs données.

5 une petite liste

Ecrivez une fonction qui transforme une string arbitrairement grande en liste.
Dans une première version on effectuera le travail grossièrement.
Dans un deuxième temps, on n'oubliera pas de traiter la ponctuation.
On pourra tester la fonction avec le petit texte de Victor Hugo, la conscience (il se trouve dans le fichier, laconscience.txt).

6 les map