Université de Pau et des Pays de l'Adour **Année 2017-2018**
Département de Mathématiques

L2-Info–Calcul scientifique
TD n^o 8

Exercice 1
Créer une classe contenant des outils permettant de calculer une racine d'une fonction f par la méthode de la sécante, dichotomie et Newton. Vous avez toute la liberté de concevoir cette classe comme bon vous semble. La seule contrainte imposée est d'écrire un main permettant de résoudre avec les outils présents dans votre classe, les quatre problèmes ci-dessous:

1. Déterminer par dichotomie la racine de $f(x) = x^3 + 2x - 2$ sur $[0, 1]$.
2. En prenant $a = 1.5$ et $b = 2$, déterminer une racine de la fonction $f(x) = \frac{1}{2}x^2 - \sin(x)$ par la méthode de la sécante.
3. Résoudre sur $\mathbb{R}^{+*}$, par la méthode de Newton On considère l'équation $-\ln(x) = x$.
4. Par la méthode de Newton, déterminer le minimum de la fonction $f(x) = \frac{1}{2}x^2 + e^{-x}$ sur $\mathbb{R}$.

Exercice 2
Créer une classe "`VecteurChar`" permettant de créer et de manipuler des vecteurs de lettres de l'alphabet et notamment de les crypter et décrypter à l'aide d'une clé. On supposera que les lettres sont exclusivement des lettres en minuscule c'est-à-dire dont le code ascii est compris entre 97 et 122. Le code ascii 97 correspond à "a" et 122 à "z". Le crytage consiste simplement à un décalage de chaque lettre à l'aide de la clé. La position dans l'alphabet de chaque lettre de la clé donne le décalage à effectuer. Chaque lettre du code est utilisée à tour de rôle pour crypter une lettre du vecteur.
Cette classe sera définie à l'intérieur d'un espace de nommage "L2Info" et devra contenir (au moins):
Les attributs : (qui devront être définis comme privé)
`int _nelem`; // longueur du vecteur
`char *_v`;// vecteur dynamique de cara
`std::string _key`; // clé de codage
`bool _encoded`; // indique si le vecteur est crypté ou pas
Les constructeurs:
`VecteurChar ()`; //constructeur sans paramètre
`VecteurChar(int n)`; //constructeur prenant la taille du vecteur en argument
`VecteurChar(const std::string& s)`; //constructeur prenant un string en argument et permettant d'initialiser le vecteur à l'aide des lettres de s
Le destructeur:
`~VecteurChar()`; //destructeur permettant de libérer la mémoire allouer pour le vecteur _v.
Les fonctions membres:
`void reDim(int n)`;//permettant reinitialiser la taille du vecteur
`int getSize() const` ; //renvoyant la taille du vecteur
`void setKey(const std::string& s)`; //permettant d'initialiser la clé de cryptage. Attention ici, lorsque la clé est changée et si le vecteur est crypter, il faudra le décrypter puis le recrypter avec la nouvelle clé.

`void encode();` // permettant de crypter le vecteur
`void decode();` //permettant de décrypter

Les opérateur interne

`char & operator [] (int i);`// accés aux éléments du vecteur
`char operator [] (int i) const;`//accés aux éléments du vecteur
`VecteurChar& operator=(const VecteurChar& v);`//opérateur de recopie
`VecteurChar& operator=(const std::string& s);`// permet d'initialiser le vecteur à partir d'un string (cf. constructeur VecteurChar(const std::string& s);)

Un opérateur externe à la classe:

`std::ostream& operator<<(std::ostream& os, const L2Info::VecteurChar& v);`//opérateur de sortie écran et fichier d'un objet de type VecteurChar

Pour exemple, le main:

```
#include "vecteurchar.h"
using namespace L2Info;
int main() {
      VecteurChar u;
      u="attentionvousetesfilmes";
      std::cerr<<u<<'\n';
      u.setKey("test");
      u.encode();
      std::cout<<u<<'\n';
      u.decode();
      std::cout<<u<<'\n';
}
```

produit la sortie:

```
attentionvousetesfilmes
txlxgxahgzgnlilxljaefik
attentionvousetesfilmes
```