

Le mardi 14 octobre 2008
de 13h30 à 16h20

Professeur : Gabriel Girard

- Seul votre manuel et les transparents du cours sont permis.
- Répondez dans le cahier prévu à cet effet.
- La correction est, entre autres, basée sur le fait que chacune de vos réponses est :
 - claire, c'est-à-dire lisible et compréhensible pour le correcteur ;
 - précise, c'est-à-dire exacte ou sans erreur ;
 - complète, c'est-à-dire que toutes les étapes de résolution du problème sont présentes ;
 - concise, c'est-à-dire que la méthode de résolution est la plus courte possible.

Question	Barème
1	/ 40
2	/ 10
3	/ 20
4	/ 30
total :	/100

1. (40 points) Les copies de sécurité

On vous demande d'écrire un programme qui implante la prise de copies de sécurité (backup) sur un ordinateur. Ce programme prend des fichiers en entrée et les copie sur un ou plusieurs médiums amovibles (CD-ROM, DVD ou ruban magnétique). Comme il y a plusieurs médiums différents, l'utilisateur spécifie en premier la taille des médiums utilisés (une seule taille pour chaque prise de copies de sécurité). À la fin, le programme affiche le nombre de médiums utilisés pour la prise de copies de sécurité.

Afin d'optimiser l'espace utilisé sur le médium, le programme doit permettre de regrouper et de compresser des fichiers.

Le regroupement de fichiers permet de sauver de l'espace dans le répertoire du médium. En effet, pour chaque fichier sauvegardé sur le médium, on copie de l'information dans le répertoire du médium qui permet de localiser ce fichier. Ainsi, chaque fichier individuel occupe 10 unités d'espace dans le répertoire du médium en plus de l'espace qu'il occupe sur le médium lui-même. Chaque regroupement de N fichiers occupe $10 + N$ unités dans le répertoire du médium en plus de l'espace qu'occupe le regroupement sur le médium. Un regroupement de fichier occupe 5 % plus d'espace que la somme des tailles des fichiers.

La compression permet aussi de sauver de l'espace. Votre programme utilise un outil de compression existant dont l'efficacité varie selon le type de fichier. La compression est de :

- 60 % si le fichier est de type «texte»(t) ;
- 10 % si la fichier est de type «son»(s) ;
- 20 % si le fichier est de type «binaire»(b) ;
- 5 % si le fichier est de type «image»(i).

Le programme fonctionne donc de la façon suivante. Il demande d'abord la taille des médiums. Ensuite, il doit, de façon répétitive, créer des groupes et les emmagasiner sur le médium. Lorsque le médium est plein, on le change et on ajuste le nombre de médiums utilisés pour la prise de copies de sécurité.

Avant de créer le groupe, le nombre de fichiers (N) à intégrer dans le groupe doit être connu. Lorsque ce nombre est connu, on lit, pour chaque fichier, son nom, son type et sa taille, on le compresse et on l'ajoute au groupe grâce à un outil de concaténation existant. Une taille négative termine la prise de copies de sécurité. Un groupe de taille «un» n'est pas considéré comme un groupe, mais comme un fichier.

**Faites l'analyse et la conception de ce problème
(pas l'implantation).**

À noter :

Le diagramme structurel ne nécessite pas plus de trois modules sous le module principal.

De plus, la décomposition ne dépasse pas quatre niveaux en profondeur.

Le nombre de modules (fonctions) à décrire dans la conception ne dépasse pas cinq.

Analyse globale

Entrées

Sorties

Formules

Constantes

Diagramme structurel

2. (10 points)

Que fait le programme suivant ?

```
#include <iostream>

using namespace std;
main()
{
    const int MAX = 10;
    for (int i = 1; i < MAX; i++)
    {
        int j;
        for (j = 0; j <= i; j++)
            cout << '*';

        for ( j = 1; j <= 2; j++)
            cout << j;

        for ( j = MAX; j >= i; j--)
            cout << '+';

        cout << endl;
    }
}
```

3. (20 points)

Écrivez un programme qui lit dans un tableau une suite de caractères terminant par un «.» (et qui n'est jamais plus longue que 100 caractères). Ce programme doit compter le nombre d'occurrences des voyelles (a, e, i, o, u) et afficher le nombre d'occurrences de chacune. On vous demande finalement de copier la chaîne inversée dans un autre tableau et de l'afficher.

4. (30 points)

Un musée organise une exposition d'artefacts anciens relatant l'histoire des 50 000 dernières années de la région. L'exposition sera divisée en cinq salles. La première contiendra tous les artefacts jusqu'à environ l'an -2 000 avant J.-C. La seconde contiendra tous les artefacts jusqu'à environ l'an -5 000 avant J.-C. La troisième contiendra tous les artefacts jusqu'à environ l'an -15 000 avant J.-C. La quatrième contiendra tous les artefacts jusqu'à environ l'an -30 000 avant J.-C. Finalement, la cinquième contiendra tous les artefacts jusqu'à environ l'an -50 000 avant J.-C.

Pour parvenir à faire son exposition, le musée se doit de déterminer l'âge de certains artefacts et de valider l'âge de d'autres (afin d'éviter les erreurs et s'assurer qu'ils sont authentiques).

Pour déterminer et valider l'âge de ses artefacts, le musée utilise la datation au carbone-14 (C14). Le musée possède l'outillage nécessaire pour mesurer le niveau de C14 dans les artefacts.

Le musée désire obtenir un programme qui lui permettra de valider et déterminer l'âge de ses artefacts. Le programme fonctionnera de la façon suivante. Il demande d'abord si on veut faire une datation (dat) ou une validation (val). Pour terminer le programme, on entre la chaîne de caractères «fin». Le programme demande ensuite les informations pertinentes pour faire la datation ou sa validation. Pour une datation, le programme requiert le niveau de C14 initial (qui a été estimé) et le niveau de C14 courant (qui a été mesuré). Il détermine ensuite dans quelle salle d'exposition on devrait retrouver l'artefact en question. Pour une validation, le programme requiert le niveau de C14 initial, le niveau de C14 courant (mesuré) et l'âge supposé de l'artefact. Il indique alors si l'artefact est authentique. Un artefact est considéré authentique si le niveau de C14 mesuré est contenu dans l'intervalle de $\pm 10\%$ de la valeur calculée.

Pour faire de la datation au carbone 14, on utilise la formule suivante :

$$A = A_0 e^{-kt}$$

où :

- A est la quantité de carbone 14 dans l'artefact après t années ;

- A_0 est la quantité originale de carbone 14 dans l'artefact ;
- e est la base du logarithme naturel ;
- k est la constante $\ln 2 / 5700$ où 5700 est la demi-vie du carbone 14 ($\ln 2 = 0.693$).

Vous devez implanter seulement les modules

- (a) «musée»,
- (b) «validation»,
- (c) «datation» et
- (d) «salle».

Les autres modules sont implantés à la page 13 du présent document.

Analyse globale du problème

Entrées :

- (clavier) Suite de types de traitement (chaînes de caractères)
- (clavier) Suite de niveaux de C14 courants (réels)
- (clavier) Suite de niveaux de C14 initiaux (réels)
- (clavier) Suite d'âges (réels)

Sorties :

- (écran) Suite d'âges (réels)
- (écran) Suite de quantités C14 calculées (réels)
- (écran) Suite de salles (entiers 1, 2, 3, 4 ou 5)
- (écran) Suite de résultats sur l'authenticité (chaînes de caractères)

Constantes :

- Code de fin de traitement = fin
- Code validation = val
- Code datation = dat
- Demi-vie carbone = 5700 ans
- $\ln(2) = 0.693$
- Marge erreur = 10 %
- Âge salle 1 = 4000 ans (l'âge des artefacts entre l'an 2000 et l'an -2000)
- Âge salle 2 = 7 000 ans
- Âge salle 3 = 17 000 ans
- Âge salle 4 = 32 000 ans
- Âge salle 5 = 52 000 ans

Formules :

- (a) $A = A_0 e^{-kt}$.
- (b) $t = (\ln(A/A_0) / -\ln(2)) \times \text{Demi-vie carbone}$
- (c) Erreur Max âge = C14 mesuré \times Marge erreur
- (d) Erreur réelle = |C14 calculé – C14 mesuré|
- (e) Pour trouver la valeur absolue, vous pouvez utiliser la fonction `fabs()` de la bibliothèque `<cmath>`.
- (f) Pour trouver le logarithme naturel (\ln), vous pouvez utiliser la fonction `log()` de la bibliothèque `<cmath>`.

Diagramme structurel

Module musee

Analyse

Entrée : (clavier) Type de traitement (caractère).

Sortie :

Constantes :

- Fin traitement = fin
- Validation = val
- Datation = dat

Conception

- (a) Pour tous les artefacts
 - i. Lecture type (val, dat ou fin)
 - ii. Selon le type de traitement (val ou dat)
 - A. Faire la validation (module *validation*)
 - B. Faire la datation (module *datation*).

Module validation

Entrées :

- (clavier) Niveau de carbone 14 courant (réel)
- (clavier) Niveau de carbone 14 initial (réel)
- (clavier) Âge présumé de l'artefact (réel)

Sorties :

- (écran) Niveau de carbone 14 calculé (réel)
- (écran) Artefact authentique ou non (Chaîne de caractères)

Constante : Marge erreur = 10 %

Formules :

- (a) Erreur maximale âge = $C14 \text{ mesuré} \times \text{Marge erreur}$
- (b) Erreur réelle âge = $|C14 \text{ calculé} - C14 \text{ mesuré}|$

Conception

- (a) Lecture
 - i. Du niveau de carbone 14 courant mesuré (c14-c)
 - ii. Du niveau de carbone 14 initial (c14-i)
 - iii. De l'âge supposé de l'artefact
- (b) Calculer le niveau de carbone 14 (module car14)
- (c) Effectuer la validation
 - i. Calculer l'erreur maximale sur l'âge (formule a)
 - ii. Calculer l'erreur réelle sur l'âge (formule b)
 - iii. Valider l'âge
- (d) Afficher le résultat de la validation

Module datation

Entrées :

- (clavier) Niveau de carbone 14 courant (réel)
- (clavier) Niveau de carbone 14 initial (réel)

Sorties :

- (écran) Âge de l'artefact (réel)
- (écran) Salle (entier)

Conception

- (a) Lecture
 - i. Du niveau de carbone 14 courant (c14-c)
 - ii. Du niveau de carbone 14 initial (c14-i)
- (b) Calculer l'âge de l'artefact (module *age*)
- (c) Déterminer la salle (module *salle*)
- (d) Afficher la salle

Module car14

Entrées :

- (paramètre) Niveau de carbone 14 initial (réel)
- (paramètre) Âge présumé de l'artefact (réel)

Sortie :

- (retour) Niveau de carbone 14 calculé (réel)

Constantes :

- Demi-vie carbone = 5 700 ans
- $\ln(2) = 0.693$

Formule : $A = A_0 e^{-kt}$

Conception

- (a) Calculer le niveau de carbone 14 (formule)

Module age

Entrées :

- (paramètre) Niveau de carbone 14 courant (réel)
- (paramètre) Niveau de carbone 14 initial (réel)

Sorties :

- (retour) Âge de l'artefact (réel)

Constantes :

- Demi-vie carbone = 5 700 ans
- $\ln(2) = 0.693$

Formule : $t = (\ln(A/A_0) / -\ln(2)) \times \text{Demi-vie carbone}$

Conception

- Calculer l'âge de l'artefact (formule)

Module salle

Entrée :

- (paramètre) Âge de l'artefact (réel)

Sortie :

- (retour) Numéro de la salle (entier)

Constantes :

- Âge salle 1 = 4 000 ans
- Âge salle 2 = 7 000 ans
- Âge salle 3 = 17 000 ans
- Âge salle 4 = 32 000 ans
- Âge salle 5 = 52 000 ans

Conception

- Selon l'âge de l'artefact
 - Choisir la bonne salle

```

/*****
* Module    car14
*
* Ce module calcule le niveau de carbone 14 a partir de
* l'age et du niveau de carbone 14 initial.
*
* Entrees : niveau de C14 initial (float)
*           age de l'artefact (float)
* Sortie  : niveau de carbone 14 courant calcule (float)
*
*****/
float car14(float c14Init, float age)
{
    float c14Calcule;

    c14Calcule = c14Init * exp(-age * 0.693/5700);
    return c14Calcule;
}

/*****
* Module    age
*
* Ce module calcule l'age d'un artefact a partir des
* niveaux de carbone 14 initial et courant.
*
* Entrees : niveau de C14 initial (float)
*           niveau de C14 courant (float)
* Sortie  : age de l'artefact (float)
*
*****/
float age(float c14Init, float c14mesure)
{
    float ageCalcule;

    ageCalcule = (log(c14mesure/c14Init) / -0.693) * 5700;
    return ageCalcule;
}

```