

Examen intermédiaire, Programmation I

Sciences et Technologies du Vivant, Semestre 1

Mercredi 4 novembre 2009

1. N'oubliez pas de mettre vos NOM et PRÉNOM sur toutes vos copies.
2. Merci de garder la donnée.
3. Vous pouvez rédiger vos réponses au crayon. Cependant, prenez soin d'écrire proprement. Les copies illisibles ne recevront pas de points.
4. Vous n'avez pas à écrire un programme entier avec `#include`, etc. Rédigez uniquement la partie demandée. Dans le cas d'une fonction, écrivez l'en-tête et le corps de la fonction uniquement.
5. Pour l'exercice 3, le nombre de lignes de code entre crochets [] au début des questions est donné à titre indicatif. Vous n'êtes pas obligés de fournir une réponse comportant *exactement* le même nombre de lignes. Par contre, si votre réponse est *beaucoup plus longue* que celle du corrigé, il est certainement possible de faire plus simple.
6. Les réponses qui font ce qui est demandé mais qui sont trop compliquées n'obtiendront pas le maximum des points.
7. Les nombres de points pour chaque question sont donnés dans la marge de droite, et sont à titre indicatif. Le total de points est sur 100.

Question 1 – Commandes Unix 20 points

Pour cet exercice, on supposera que votre nom d'utilisateur est `username`. Vous avez démarré un terminal et vous trouvez dans un répertoire quelconque. Un listing de ce répertoire affiche les 9 fichiers suivants :

```
-rw-r--r-- 1 username sv-ba1 134 2009-10-26 20:55 file1-04.txt
-rw-rw-rw- 2 username sv-ba1 6251 2009-10-26 20:31 file2.txt
-rw-r--r-- 1 username sv-ba1 3192 2009-10-26 20:56 file3-9.doc
-rw-r--r-- 3 username sv-ba1 4010 2009-10-26 21:14 file
-rw-rw-r-- 1 username sv-ba1 19283 2009-10-26 20:56 picture7.jpg
-rw-r--r-- 1 username sv-ba1 30291 2009-10-26 20:57 picture19.jpg
drwxr-xr-x 2 username sv-ba1 4096 2009-10-26 20:57 init.d
-rw----- 1 username sv-ba1 98182 2009-10-26 20:57 music_49.mp3
-rwxr-xr-x 1 username sv-ba1 2048 2009-10-26 20:52 ifdown
```

- (a) Parmi les fichiers listés, y a-t-il des répertoires ? Si oui, lesquels ? (2)
- (b) Entrez la commande permettant de créer un nouveau sous-répertoire appelé `texte`. (2)
- (c) En **une seule commande**, copiez les **trois premiers fichiers** du listing dans le répertoire `texte`. Faites-le sans énumérer intégralement les trois fichiers en question. (4)
- (d) Écrivez la commande nécessaire afin de supprimer le fichier `music_49.mp3`. (2)
- (e) Un de vos camarades, dont le nom d'utilisateur est `ami`, vous invite à récupérer un fichier intitulé `fichier.txt` qui se trouve dans son *home directory*. Écrivez la commande permettant de copier ce fichier et de le placer dans votre répertoire `texte`, tout en le renommant `fichier_ami.txt`. (4)

- (f) Créez un nouveau sous-répertoire `images`. Ensuite, déplacez-y les deux fichiers dont l'extension est `.jpg`. Modifiez les droits de ce nouveau répertoire, de manière à ce que les membres de votre groupe puissent non seulement lire vos images, mais également y ajouter les leurs. (6)

Question 2 – Pointeurs 38 points

- (a) Qu'affiche le programme suivant? (15)

```
int f(int n)
{
    return n + 1;
}

int main(int argc, char ** argv)
{
    int T[6] = {4, 3, 0, 2, 5, 1};

    int a = 0, b = T[a];
    cout << "A " << a << " " << b << endl;

    a = T[a];
    b = b + T[a];
    cout << "B " << a << " " << b << endl;

    do {
        a = T[a];
        if (a != 0)
            b = b + T[a];
    } while (a != 0);
    cout << "C " << a << " " << b << endl;

    cout << "D " << T[ T[ T[0]]] << " " << T[ T[ T[ T[ T[ T[0]]]]]] << endl;

    cout << "E " << f(T[f(0)]) << " " << T[f(T[0])] << endl;

    return 0;
}
```

- (b) Qu'affiche le programme suivant? (9)

```
bool T[3];
T[0] = false;
T[1] = false;
T[2] = true;

bool a = true;
for(int i = 0; i < 3; i++)
    a = a && T[i];
if (a)
    cout << "A oui" << endl;
else
    cout << "A non" << endl;

a = false;
```

```

for(int i = 0; i < 3; i++)
    a = a || T[i];
if (a)
    cout << "B oui" << endl;
else
    cout << "B non" << endl;

```

```

a = true;
for(int i = 0; i < 3; i++)
    a = a || T[i];
if (a)
    cout << "C oui" << endl;
else
    cout << "C non" << endl;

```

```

for(int i = 0; i < 3; i++)
    T[i] = !T[i];
cout << "D ";
for(int i = 0; i < 3; i++)
    if (T[i])
        cout << "0";
    else
        cout << "1";
cout << endl;

```

(c) Qu'affiche le programme suivant ?

(14)

```

int a = 17;
int *b = &a;
int *c = b;

a++;
b--;
*c -= 3;

cout << "a) " << a << " " << *c << endl;

a = 7;
int d = *c;
b = &d;
(*c)++;
(*b)--;

cout << "b) " << a << " " << d << endl;

a = 3;
d = 2;
int *t = new int[a];
c = t;
for (int i=0; i<a; i++) {
    *c = a * d - i;
    c++;
}

```

```

cout << "c) ";

for (int i=0; i<a; i++)
    cout << t[i] << " ";
cout << endl;

```

Question 3 – Mandelbrot 46 points

(a) [7 lignes] (6)

On considère la suite (U_n) paramétrée par le réel a :

$$\begin{cases} U_0 = a \\ U_n = U_{n-1}^2 + a \end{cases}$$

Écrivez le code qui demande à l'utilisateur une valeur pour a , et affiche les 10 premiers termes de la suite U_0 à U_9 .

(b) [10 lignes] (12)

Écrivez une fonction `nb_iterations` qui a comme paramètre la valeur de a , et calcule les termes de la suite jusqu'à ce que la valeur absolue de U_n devienne plus grande que 2. La fonction devra alors retourner la valeur de n . Si la valeur absolue de U_n ne devient jamais plus grande que 2 pour $n < 100$, la fonction devra retourner -1.

Plus formellement, la fonction devra retourner :

$$\begin{cases} -1 & \text{si } \forall n < 100 \ |U_n| < 2, \text{ et} \\ n \text{ tel que } |U_n| \geq 2 \text{ et } \forall p < n, \ |U_p| < 2 & \text{sinon.} \end{cases}$$

La fonction C pour calculer la valeur absolue est `float fabs(float x)`.

(c) [2 lignes] (8)

Vous pouvez répondre à cette question même si vous n'avez pas su répondre à la question précédente.

Écrivez le code qui affiche

$$\begin{cases} -1 & \text{si } \forall n < 100 \ |U_n| < 2, \text{ et} \\ n \text{ tel que } |U_n| \geq 2 \text{ et } \forall p < n, \ |U_p| < 2 & \text{sinon,} \end{cases}$$

pour $a = [0, 0.01, 0.02, 0.03 \dots 1]$. Vous pouvez naturellement avoir recours à la fonction `nb_iterations` implémentée au point précédent. L'affichage devra ressembler à :

```

a = 0 -> -1
a = 0.01 -> -1
...
a = 0.26 -> 29
a = 0.27 -> 19
...
a = 0.99 -> 2
a = 1 -> 1

```

(d) [13 lignes] (20)

On veut calculer maintenant les premiers termes de la suite *complexe* (Z_n)

$$\begin{cases} Z_0 = c \\ Z_n = Z_{n-1}^2 + c \end{cases}$$

Cette fois, c est un nombre complexe, et (Z_n) prend donc ses valeurs dans l'ensemble des nombres complexes. Une façon de calculer les termes de (Z_n) est de calculer simultanément deux suites réelles, une suite (R_n) contenant la partie réelle de (Z_n) , et une suite (I_n) contenant la partie imaginaire. On a donc ici :

$$\begin{cases} R_0 = c_{re}, & I_0 = c_{im} \\ R_n = R_{n-1}^2 - I_{n-1}^2 + c_{re}, & I_n = 2R_{n-1}I_{n-1} + c_{im}, \end{cases}$$

où c_{re} et c_{im} sont les parties réelle et imaginaire de c .

Écrivez le code qui demande à l'utilisateur c_{re} et c_{im} , et affiche les 10 premiers termes de la suite (Z_n) pour ces valeurs. L'affichage devra donc ressembler à :

```
Entrez la valeur de c_re
0.1
Entrez la valeur de c_im
0.2
Z1 = 0.07 + i 0.24
Z2 = 0.0473 + i 0.2336
Z3 = 0.0476683 + i 0.222099
Z4 = 0.0529445 + i 0.221174
Z5 = 0.0538851 + i 0.22342
Z6 = 0.0529872 + i 0.224078
Z7 = 0.0525967 + i 0.223747
Z8 = 0.0527039 + i 0.223537
Z9 = 0.0528091 + i 0.223563
Z10 = 0.0528086 + i 0.223612
```