

# Examen intermédiaire, Programmation I

Sciences et Technologies du Vivant, Semestre 1

Mercredi 9 novembre 2011

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. Les réponses qui font ce qui est demandé mais qui sont trop compliquées n'obtiendront pas le maximum des points.
6. 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 ..... 15 points

Supposons que le répertoire courant est un répertoire appelé `frigo`. Quand on exécute la commande `ls -al`, on obtient l'affichage suivant :

```
drwxr-xr-x  2 lepetit cvlab  102 Oct 28 11:32 .
drwxr-xr-x  2 lepetit cvlab  442 Oct 28 11:32 ..
drwxr-xr-x  2 lepetit cvlab 4096 Oct 28 11:32 chanteuses
drwxr-xr-x  2 lepetit cvlab 4096 Oct 28 11:32 fruits
drwxr-xr-x  2 lepetit cvlab 4096 Oct 28 11:32 legumes
```

Quand on exécute la commande `ls`, on obtient l'affichage suivant :

```
chanteuses fruits legumes
```

Quand on exécute la commande `ls chanteuses`, on obtient l'affichage suivant :

```
britney justin ladygaga rihanna
```

Quand on exécute la commande `ls fruits`, on obtient l'affichage suivant :

```
ananas fraises legumes poires pois pommes
```

Quand on exécute la commande `ls legumes`, on obtient l'affichage suivant :

```
poireaux tomates
```

Quand on exécute la commande `ls fruits/legumes`, rien ne s'affiche.

- (a) Supposons qu'on entre les commandes suivantes : (3)

```
mv legumes/tomates fruits/legumes
ls legumes
```

Que va-t-il s'afficher pour la dernière commande ?

- (b) Supposons qu'on entre les commandes suivantes : (3)

```
mv fruits/poi?*s legumes
ls legumes
```

- Que va-t-il s'afficher pour la dernière commande?
- (c) Supposons qu'on entre les commandes suivantes, et que `fraises` soit un fichier : (3)
- ```
cp fruits/fraises ..
ls
```
- Que va-t-il s'afficher pour la dernière commande?
- (d) Supposons qu'on entre les commandes suivantes : (3)
- ```
ls legumes/../../fruits/./legumes/../../..
```
- Que va-t-il s'afficher pour la dernière commande?
- (e) Comment déplacer les fichiers `ladygaga` et `rihanna` dans le répertoire `legumes` en une seule commande? (3)

## Question 2 – Compréhension de code ..... 25+30=55 points

- (a) Indiquez ce qu'affiche le code suivant : (3+3+4+4+5+6=25)

```
int T[5] = {1, 2, 3, 4, 5};
```

```
cout << "A)" << endl;
for(int i = 4; i > 0; i--)
    cout << T[i] << " ";
cout << endl;
```

```
cout << "B)" << endl;
int * p = T;
for(int i = 4; i > 0; i--)
{
    cout << *p << " ";
    p = p + 1;
}
cout << endl;
```

```
cout << "C)" << endl;
p = T;
for(int i = 0; i < 5; i++)
{
    cout << *p << " ";
    if (i % 2 == 0)
        p = p + 1;
}
cout << endl;
```

```
cout << "D)" << endl;
p = T;
for(int i = 0; i < 5; i++)
{
    cout << *(p + 4 - i) << " ";
}
cout << endl;
```

```
cout << "E)" << endl;
p = T;
for(int i = 0; i < 5; i++)
```

```

    {
        cout << *(p + 4 - i) << " ";
        p = p + 1;
    }
cout << endl;

cout << "F)" << endl;
for(int i = 0; i < 5; i++)
    T[i] = i;
p = T;
int * q = p;
for(int i = 0; i < 5; i++)
    {
        cout << *p << " ";
        p = p + 1;
        *q = *q + 1;
    }
cout << endl;

```

(b) Indiquez ce qu'affiche le code suivant :

(3+3+3+3+6+6+6=30)

```

#include <iostream>
using namespace std;

const int N = 5;

void set_to_0(bool * t)
{
    for(int i = 0; i < N; i++)
        t[i] = false;
}

void aff(bool * t)
{
    for(int i = N-1; i >= 0; i--)
        if (t[i])
            cout << "1";
        else
            cout << "0";

    cout << endl;
}

void f(bool * rs, bool * t, bool * u)
{
    for(int i = 0; i < N; i++)
        rs[i] = t[i] && u[i];
}

void g(bool * rs, bool * t, bool * u)
{
    for(int i = 0; i < N; i++)
        rs[i] = t[i] || u[i];
}

```

```

void h(bool * t)
{
    if (t[0] == true)
        {
            t[0] = false;
            t[1] = true;
        }
    else
        t[0] = true;
}

void k(bool * t)
{
    bool retenue = true;
    t[0] = (t[0] != retenue);
    retenue = !t[0];
    t[1] = retenue;
}

void m(bool * t)
{
    int i = 0;

    while(t[i])
        {
            t[i] = false;
            i++;
        }
    t[i] = true;
}

int main(int argc, char ** argv)
{
    bool a[N], b[N], c[N];

    cout << "A ";
    for(int i = 0; i < N; i++)
        a[i] = (i % 2 == 0);
    aff(a);

    cout << "B ";
    for(int i = 0; i < N; i++)
        b[i] = !a[i];
    aff(b);

    cout << "C ";
    f(c, a, b);
    aff(c);

    cout << "D ";
}

```

```

g(c, a, b);
aff(c);

cout << "E) " << endl;
set_to_0(a);
h(a);
cout << "1: ";
aff(a);
h(a);
cout << "2: ";
aff(a);
h(a);
cout << "3: ";
aff(a);

cout << "F) " << endl;
set_to_0(a);
k(a);
cout << "1: ";
aff(a);
k(a);
cout << "2: ";
aff(a);
k(a);
cout << "3: ";
aff(a);

cout << "G) " << endl;
set_to_0(a);
for(int i = 1; i <= 3; i++) {
    m(a);
    cout << i << ": ";
    aff(a);
}

return 0;
}

```

**Question 3** – Motif ..... 30 points

On vous donne les deux fonctions suivantes :

```

void t(int n)
{
    for(int i = 0; i < n; i++)
        cout << "*";
}

```

```

void s(int n)
{
    for(int i = 0; i < n; i++)
        cout << " ";
}

```

- (a) Utilisez ces deux fonctions pour écrire une fonction qui affiche une ligne, similaire à la ligne : (10)

```
****      ****
```

Le nombre d'étoiles à droite et à gauche devra être le même, et défini comme un paramètre à la fonction. Le nombre d'espaces entre les étoiles sera également défini comme un paramètre.

- (b) Utilisez la fonction que vous venez d'écrire pour afficher le motif suivant : (20)

```
**** ****
***  ***
**   **
*    *

*      *
**     **
***    ***
****   ****
```

La fonction devra avoir un paramètre permettant de définir le nombre de lignes du motif. On supposera que ce paramètre est toujours impair et supérieur ou égal à 3 quand on appelle la fonction.