

Corrigé de l'examen facultatif du 13 décembre 2006
Programmation I
Sciences et Technologies du Vivant, Semestre 1
Chimie et Génie Chimique, Semestre 3

Pour chaque question, il y a souvent plusieurs façons plus ou moins équivalentes de répondre. Voici les réponses que je considère comme les meilleures.

Question 1 - Commandes Unix

- (a) `man find`
- (b) `chmod u+w rep`
`cp fichier.txt rep`
- (c) `mkdir rep2/perso`
`chmod o-rwx rep2/perso`
- (d) `mv rep/fichier.txt rep2/perso`
- (e) `mv rep2/perso rep2/personnel`

Question 2 - Fonctions et Booléens

1.

```
bool premiers_entre_eux(int a, int b)
{
    return pgcd(a, b) == 1;
}
```
2.

```
bool premiers_entre_eux(int a, int b, int c)
{
    return pgcd(pgcd(a, b), c) == 1;
}
```
3.

```
bool premiers_entre_eux(int a, int b, int c)
{
    if (pgcd(a, b) == 1)
        return true;
    else
        return pgcd(pgcd(a, b), c) == 1;
}
```

ou mieux :

```
bool premiers_entre_eux(int a, int b, int c)
{
    int pgcd_a_et_b = pgcd(a, b);

    if (pgcd_a_et_b == 1)
```

```

        return true;
    else
        return pgcd(pgcd_a_et_b, c) == 1;
    }
4. bool multiple(int a, int b)
{
    int t = 1;

    while(a > b * t)
        t++;

    return a == b * t;
}
5. bool divide(int x, int y, int z)
{
    if (z == 0)
        return false;

    if (x == y * z)
        return true;

    return (x, y, z - 1);
}

bool multiple(int a, int b)
{
    return divide(a, b, a);
}

```

Question 3 - Tableaux

```

(a) int *alloue_tableau(int n) {
    int *t = new int[n+1];
    t[0] = n;
    return t;
}
(b) void libere_tableau(int *t) {
    delete[] t;
}
(c) int taille(int *t) {
    return t[0];
}
(d) int lire(int *t, int pos) {
    if (pos < 0 || pos >= taille(t))
        return 0;
    else
        return t[pos+1];
}

```

```
(e) int somme(int *t) {
    int s = 0;
    for (int i=0; i<taille(t); i++)
        s += lire(t, i);
    return s;
}
```

(f) Solution 1 :

```
int *fusion(int *t1, int *t2) {
    int *fus = alloue_tableau(taille(t1) + taille(t2));

    int index1 = 0;
    int index2 = 0;
    int index_fus = 0;

    while (index1 < taille(t1) && index2 < taille(t2)) {
        if (lire(t1, index1) < lire(t2, index2)) {
            ecrire(fus, index_fus, lire(t1, index1));
            index1++;
        }
        else {
            ecrire(fus, index_fus, lire(t2, index2));
            index2++;
        }
        index_fus++;
    }

    if (index1 < taille(t1)) {
        while (index1 < taille(t1)) {
            ecrire(fus, index_fus, lire(t1, index1));
            index1++;
            index_fus++;
        }
    }
    if (index2 < taille(t2)) {
        while (index2 < taille(t2)) {
            ecrire(fus, index_fus, lire(t2, index2));
            index2++;
            index_fus++;
        }
    }

    return fus;
}
```

Solution 2 :

```
int *fusion(int *t1, int *t2) {
    int *fus = alloue_tableau(taille(t1) + taille(t2));
```

```

int index1 = 0;
int index2 = 0;
int index_fus = 0;

for (int i=0; i<taille(t2); i++) {
    while (lire(t1, index1) < lire(t2, i) && index1 < taille(t1)) {
        ecrire(fus, index_fus, lire(t1, index1));
        index1++;
        index_fus++;
    }
    ecrire(fus, index_fus, lire(t2, i));
    index_fus++;
}
return fus;
}

```

Question 4 - Pointeurs

	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	...
(a)	28	36						0	2	20	4	8				
	t	n														

- (b)
1. 34
 2. i: 34, j: 17
 3. i: 24, j: 12
 4. i: 24, j: 5, q: 24
- (c)
1. 11
 2. 4
 2. 0
 3. 2
- Segmentation fault