

Corrigé de l'examen intermédiaire, Programmation I  
Sciences et Technologies du Vivant, Semestre 1

Mercredi 14 novembre 2007

**Question 1 - Commandes Unix**

- (a) `ls -l temp/a*.avi temp/b*.avi`
- (b) `mkdir videos images`
- (c) `mv temp/*.avi videos/`  
`mv temp/*.jpg images/`
- (d) `rm -r temp`
- (e) `chmod go-x images`

**Question 2 - Pointeurs**

- (a) 1. 10 1  
2. 12 5  
3. 2 1  
4. 1 3 1 7 8
- (b) 1. 4 6  
2. 6  
3. 5  
4. 8
- (c) 1. 5  
2. 5  
3. 22  
4. 13

**Question 3 - Implantation de racines**

- (a) 

```
float mon_sqrt(float x, int n) {
    float ui = x;
    for (int i=0; i<n; i++)
        ui = 0.5 * (ui + x / ui);
    return ui;
}
```

```

(b) float *cree_tableau_pour_sqrt(int m) {
    float *tab = new float[m];
    for (int i=0; i<m; i++)
        tab[i] = mon_sqrt(i, 10);
    return tab;
}

(c) float mon_sqrt_avec_tableau_precalcule(float x, float *tab, int m) {
    if (x < m)
        return 0.5 * (tab[int(x)] + x / tab[int(x)]);
    else
        return mon_sqrt(x, 10);
}

(d) float mon_sinus(float x, int n) {
    float sum = x;
    float term = x;
    for (int i=1; i<n; i++) {
        term = term * (-1) * x * x / ( (2 * i) * (2 * i + 1) );
        sum += term;
    }
    return sum;
}

(e) void trouve_m_et_y(float x, float *y, int *m) {
    *m = int(x * 2 / M_PI);
    *y = x - (*m * M_PI / 2);
}

(f) float mon_sinus2(float x) {
    int m;
    float y;
    trouve_m_et_y(x, &y, &m);
    switch (m % 4) {
    case 0:
        return mon_sinus(y, 10);
        break;
    case 1:
        return mon_cosinus(y, 10);
        break;
    case 2:
        return -mon_sinus(y, 10);
        break;
    case 3:
        return -mon_cosinus(y, 10);
}

```

```

        break;
    }
}

```

Également possible:

```

float mon_sinus2(float x) {
    int m;
    float y;
    trouve_m_et_y(x, &y, &m);
    if (m % 4 == 0)
        return mon_sinus(y, 10);
    else if (m % 4 == 1)
        return mon_cosinus(y, 10);
    else if (m % 4 == 2)
        return -mon_sinus(y, 10);
    else if (m % 4 == 3)
        return -mon_cosinus(y, 10);
}

```

#### Question 4 - Tableaux

- (a) 

```
int *copie_tableau(int *tab, int taille) {
    int *nouveau = new int[taille];
    for (int i=0; i<taille; i++)
        nouveau[i] = tab[i];
    return nouveau;
}
```
- (b) 

```
void melange_tableau(int *tab, int taille, int nbr_permutations) {
    for (int i=0; i<nbr_permutations; i++) {
        // tire les elements a permuter
        int e1 = rand() % taille;
        int e2 = rand() % taille;

        // effectue la permutation
        int temp = tab[e1];
        tab[e1] = tab[e2];
        tab[e2] = temp;
    }
}
```
- (c) 

```
bool alphabetique(char *c1, char *c2) {
    int curseur = 0;
    int len_min;
    int l1 = strlen(c1);
```

```
int l2 = strlen(c2);
if (l1 < l2)
    len_min = l1;
else
    len_min = l2;
while (curseur < len_min) {
    if (c1 < c2)
        return true;
    else
        if (c1 > c2)
            return false;
    curseur++;
}
if (l1 < l2)
    return true;
else
    return false;
}
```