

Corrigé de l'examen facultatif du 14 décembre 2005
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 - Manipulation de tableaux

(a)

```
int S[nb_elements];
for (int i=0; i<nb_elements; i++) {
    S[i] = 0;
    for (int j=0; j<=i; j++)
        S[i] += T[j];
}
```

ou :

```
int S[nb_elements];

S[0] = T[0];
for(int i = 1; i < nb_elements; i++)
    S[i] = S[i-1] + T[i];
```

(b)

```
int P[nb_elements];
for (int i=0; i<nb_elements; i++) {
    P[i] = 0;
    for (int j=0; j<=i; j++)
        if (T[j] % 2 == 0)
            P[i]++;
}
```

OU

```
int P[nb_elements];
if (T[0] % 2 == 0)
    P[0] = 1;
else
    P[0] = 0;
for(int i = 1; i < nb_elements; i++) {
```

```

    if (T[j] % 2 == 0)
        P[i] = P[i-1] + 1;
    else
        P[i] = P[i-1];
}

```

Question 2 - Tableaux de booléens

(a)

```

for(int i = 0; i < nb_elements; i++)
    T[i] = !T[i];

```

(b)

```

bool au_moins_un_vrai = false;
int i = 0;
while (!au_moins_un_vrai && i < nb_elements)
{
    if (T[i])
        au_moins_un_vrai = true;

    i++;
}
if (au_moins_un_vrai)
    cout << "Au moins 1 des elements est a 'vrai'" << endl;

```

Question 3 - Puissances

(a)

```

float puissance(float a, int n) {
    float resultat = 1;

    for (int i = 0; i < n; i++)
        resultat = resultat * a;

    return resultat;
}

```

(b)

```

float puissance_recursive(float a, int n)
{
    if (n == 0)
        return 1;
    else if (n % 2 == 0)
    {
        float t = puissance_recursive(a, n / 2);

        return t * t;
    }
}

```

```

else
{
    float t = puissance_recursive(a, (n - 1) / 2);

    return t * t * a;
}
}

```

Question 4 - Figures graphiques

(a)

```

void affiche_ligne(int largeur, int etoiles) {
    int espaces = (largeur - etoiles) / 2;

    for (int i = 0; i < espaces; i++)
        cout << " ";
    for (int i = 0; i < etoiles; i++)
        cout << "*";
    cout << endl;
}

```

(b)

```

void affiche_trapeze(int pb, int gb, int largeur) {
    for (int i = pb; i <= gb; i = i + 2)
        affiche_ligne(largeur, i);
}

```

(c)

```

affiche_trapeze(1, 5, 9);
affiche_trapeze(3, 7, 9);
affiche_trapeze(5, 9, 9);
affiche_trapeze(3, 3, 9);

```

Question 5 - Triangle de Pascal

```

void triangle_pascal(int hauteur) {
    int ligne_courante[100];
    int ligne_precedente[100];

    for (int ligne=0; ligne<hauteur; ligne++) {
        for (int col=0; col<=ligne; col++) {
            if (col == ligne) {
                ligne_courante[col] = 1;
                cout << "1" << endl;
            } else if (col == 0) {
                ligne_courante[col] = 1;
                cout << "1 ";
            } else {
                ligne_courante[col] = ligne_precedente[col] + ligne_precedente[col-1];
            }
        }
    }
}

```

```

        cout << ligne_courante[col] << " ";
    }
}
// recopie ligne_courante dans ligne_precedente
for (int i=0; i<hauteur; i++)
    ligne_precedente[i] = ligne_courante[i];
}
}

```

OU

```

void triangle_pascal(int hauteur)
{
    int L[100];

    for (int i = 0; i <= hauteur; i++)
    {
        L[i] = 1;
        for (int j = i - 1; j > 0; j--)
            L[j] = L[j] + L[j - 1];
        L[0] = 1;

        for(int j = 0; j <= i; j++)
            cout << L[j] << " ";
        cout << endl;
    }
}

```

Question 6 - Mémoire et variables

(a)

- a1) 5 5
- a2) 8
- a3) 8
- a4) 8

(b)

- b1) 12, 6
- b2) 12, 24
- b3) 3, 6
- b4) 12, 24

(c)

- c1) 9
 - c2) 10
 - c3) 20
- Segmentation fault