

Corrigé de l'examen 2007, Programmation I
Sciences et Technologies du Vivant, Semestre 1
Chimie et Génie Chimique, Semestre 3

Exercice 1 - Élevage de lapins

- ```
(a) struct Lapin {
 char *nom;
 int age;
 float poids;
 int x, y;
};

(b) struct Enclos {
 Lapin *lapins;
 int nbr_lapins;
 int longueur, largeur;
};

(c) bool interieur(Lapin *l, Enclos *e) {
 if (l->x >= 0 && l->x < e->longueur &&
 l->y >= 0 && l->y < e->largeur)
 return true;
 else
 return false;
}

(d) bool alarme(Enclos *e) {
 for (int i=0; i < e->nbr_lapins; i++)
 if (!interieur(e->lapins + i, e))
 return true;

 return false;
}

(e) void affiche_enclos(SimpleWindow *win, Enclos *enc, int marge) {
 // affiche enclos
 win->color(0.5, 0.5, 0.5);
 win->fillRectangle(marge, marge, enc->longueur, enc->largeur);

 // affiche lapins
 for (int i=0; i<enc->nbr_lapins; i++)
```

```

 affiche_lapin(win, enc->lapins[i].nom, enc->lapins[i].x + marge,
 enc->lapins[i].y + marge, interieur(enc->lapins + i, enc));
 }

```

```

(f) Lapin *plus_proche_lapin(Enclos *enc, Lapin *lap) {
 Lapin *plus_proche = 0;
 int min = 0, distance;
 for (int i=0; i<enc->nbr_lapins; i++)
 if (enc->lapins + i != lap) {
 distance = distance_lapins(enc->lapins + i, lap);
 if (distance < min || plus_proche == 0) {
 plus_proche = enc->lapins + i;
 min = distance;
 }
 }
 return plus_proche;
}

```

```

(g) struct Lapin {
 char *nom;
 int age;
 float poids;
 int x, y;
 Lapin *pere;
 Lapin *mere;
};

```

```

(h) void affiche_parents(Lapin *l) {
 cout << l->nom << " est le fils de " << l->pere->nom
 << " et " << l->mere->nom << endl;
}

```

## Exercice 2 - Cryptographie de Boy-Scout

```

(a) void codage_jules_cesar(char * phrase, int n) {
 for (int i=0; i<strlen(phrase); i++)
 if (phrase[i] != ' ')
 phrase[i] = 'A' + ((phrase[i] - 'A' + n) % 26);
}

```

```

(b) char lettre_la_plus_frequente(char *phrase) {
 int frequence[26], max = 0, lettre = 0;
 for (int i=0; i<26; i++)
 frequence[i] = 0;
 for (int i=0; i<strlen(phrase); i++)
 if (phrase[i] != ' ')
 frequence[phrase[i] - 'A']++;
 for (int i=0; i<26; i++)

```

```

 if (frequence[i] > max) {
 max = frequence[i];
 lettre = i;
 }
 return 'A' + lettre;
}

```

```

(c) void decodage_jules_cesar(char *phrase) {
 char c = lettre_la_plus_frequente(phrase);
 codage_jules_cesar(phrase, 26 + 'E' - c);
}

```

```

(d) void codage_avec_table(char * phrase, char table[128]) {
 for (int i=0; i<strlen(phrase); i++)
 if (phrase[i] != ' ')
 phrase[i] = table[int(phrase[i])];
}

```

```

(e) void decodage_avec_table(char * phrase, char table[128]) {
 for (int i=0; i<strlen(phrase); i++)
 if (phrase[i] != ' ') {
 int j = int('A');
 while (table[j] != phrase[i]) j++;
 phrase[i] = char(j);
 }
}

```

### Exercice 3 - Pointeurs

```

(a) 1a. 1 2 3
 1b. 2 1 2
 2a. 2 1 2
 2b. 2 1 2
 3. 2 1 3
 4. 1 2 1
 5a. 2 1 2
 5b. 2 1 2

```

```

(b) A: 1
 B: 2
 C: 2
 D: 1
 E: 1
 F: 3

```