

# Examen d'Algorithmique

2ème session (Juin 2010)  
Formation ingénieur CNAM

Durée : 2h

Notes de cours, Tds et Tps autorisées

---

*Pour tous les exercices nécessitant l'écriture de programmes, écrivez un code C lisible, correctement indenté, spécifié, et commenté utilement lorsque nécessaire. Utilisez des noms de variables suffisamment explicites. N'utilisez pas de stylo rouge ni de crayon à papier. Les portables doivent être éteints et rangés dans les sacs. Le barème approximatif indiqué est donné à titre indicatif et est susceptible d'être modifié.*

## **Exercice 1** (environ sur 6 pts)

On se donne les deux fonctions suivantes, avec la pré-condition  $n \geq 1$ :

```
int f1 (int n){
    int u=1, v=1;
    return f2_iter (n, &u, &v);
}

int f2_iter (int n, int* p1, int* p2){
    int* p3;
    int i;
    for(i=2; i<n; i++){
        *p1 = *p1 + *p2;
        p3=p1; p1=p2; p2=p3;
    }
    return *p2;
}
```

- Calculer  $f1(1)$ ,  $f1(2)$ ,  $f1(3)$ ,  $f1(4)$ , et  $f1(5)$ .
- Reconnaissez-vous cette fonction ?
- Concevoir et programmer une fonction récursive  $f2\_rec$  qui a le même prototype que  $f2\_iter$ , et qui fait la même chose, mais de manière récursive.

## **Exercice 2 Le Mastermind** (environ sur 14 pts)

Le *Mastermind* est un jeu de réflexion. Il se présente généralement sous la forme d'un plateau ayant 12 rangées de 4 trous pouvant accueillir des pions de couleurs. Le nombre de couleurs différentes est de 8. Avant de démarrer la partie, l'ordinateur va tirer au hasard une combinaison de 4 pions de couleurs (éventuellement identiques). Le but est de retrouver quels sont les 4 pions choisis par l'ordinateur et d'en connaître les positions.

Pour cela, à chaque tour, le joueur doit se servir des pions de couleurs pour former une rangée (une proposition). Alors à chaque tour l'ordinateur devra indiquer :

- le nombre de pions de bonne couleur bien placés.
- le nombre de pions de bonne couleur mais mal placés.

En aucun cas la position des pions bien placés est indiquée. Le joueur peut faire des propositions tant qu'il n'a pas trouvé les 4 pions choisis par l'ordinateur.

Pour simplifier, dans tout l'exercice, au lieu d'utiliser des couleurs nous utiliserons des entiers compris entre 1 et 8. La solution et les propositions de l'utilisateur seront représentées par des tableaux de 4 entiers.

Exemples : supposons que la solution est 4,2,2,1.

- Si la proposition est 2,2,1,1, il y a deux pions bien placés et un pion mal placé
- Si la proposition est 2,4,5,6, il n'y a aucun pion bien placé mais il y a deux pions mal placés
- Si la proposition est 4,2,2,6, il y a trois pions bien placés mais il n'y a aucun pion mal placé

a) Programmer une fonction `bien_places` qui prend en argument deux tableaux de quatre entiers (une proposition et la solution) et renvoie le nombre de pions bien placés.

```
int bien_places (int proposition[4],int solution[4])
```

b) Programmer une fonction `recherche_et_efface` qui possède les spécifications suivantes:

```
\\ Nom: recherche_et_efface
\\ Arguments: un entier n et un tableau de quatre entiers t.
\\ Résultat: 1 si l'entier n apparaît dans le tableau t, 0 sinon.
\\ Pré-condition: aucune
\\ Arguments modifiés en sortie: si la première occurrence de n
    apparaissait dans le tableau t à l'indice i
    alors t[i] doit valoir -1, si n n'apparaissait
    pas dans t n'est pas modifié.
```

c) Programmer une fonction `copie` qui prend en argument un tableau de 4 entiers et renvoie une copie de ce tableau.

```
int * copie(int t[4])
```

d) En déduire une fonction `mal_places` qui prend en arguments une proposition et la solution et renvoie le nombre de pions mal placés.

e) On suppose que l'on a une fonction `int hasard(int min, int max)` qui renvoie un nombre au hasard compris entre `min` et `max`. Programmer une fonction

```
int * invente()
```

qui renvoie un tableau de quatre entiers tirés au hasard entre 1 et 8.

f) Programmer une fonction `joue` qui prend en argument un tableau de quatre entiers représentant la solution. Cette fonction ne renvoie rien mais demande à l'utilisateur des propositions et affiche à l'écran le nombre de pions bien placés et mal placés tant qu'il n'a pas trouvé la solution.