

Examen de rattrapage (Durée 1h30)

Exercice 1: (6points)

- a) Voici la fonction de Morris : (2pts)

int Morris (int m, int n)

```
{ if (m == 0) return(1);
  else return(morris(m - 1, morris(m, n)));
}
```

Que vaut *Morris (1,0)* ?

- b) Soit un tableau T de n entiers. Ecrire une fonction récursive qui recherche, une valeur val donnée, par la méthode dichotomique. (4pts)

Exercice 2: (7points)

Soient deux piles¹ P1 et P2 dont chaque élément contient le coefficient et le degré d'un polynôme. **Exemple** : $p1(x) = 4x^3 - 7x + 9$; $p2(x) = 13x - 25$

4	3
0	2
-7	1
9	0

Pile P1

13	1
-25	0

Pile P2

4	3
0	2
6	1
-16	0

Pile P3

- a) Ecrire une fonction **LirePoly** qui initialise une pile **P** avec un polynôme donnée.
 b) Ecrire une fonction **SommePoly** qui retourne une pile **P3** qui contient le polynôme somme de **P1** et **P2**
 c) Ecrire le programme qui initialise deux polynômes puis calcul leur somme et l'affiche.

Exercice 3: (7 points)

Soit N un entier positif. On veut construire la liste² L des **nombre premiers** de 2 à N , comme suit: on construit la liste L des entiers de 2 à N , puis on supprime de L tous les nombres non premiers (les multiples de 2, les multiples de 3, les multiples de 5 etc....).

- a) Écrire une fonction **liste ConstListe(int N)** qui construit la liste des entiers de 2 à N .
 b) Ecrire une fonction **void SupMultiple(liste L, int K)** qui supprime de la liste L tous les multiples de K .
 c) Ecrire une fonction **liste NbPremiers(int N)** qui construit la liste des nombres premiers jusqu'à N .

¹ Les fonctions sur les piles, dont les prototypes sont donnés ci-dessous, sont supposées prédéfinies :
pile initpile() ; **int pilevide(pile p)** ; **typelem sommetpile(pile p)** ;

void empiler(pile *p, typelem x) ; **void desempiler(pile *p, typelem *x)** ;

² La fonction suivante : **liste créer_noeud()** ; ; **void ajout_tete(liste *tete, typelem E)** ;
void ajout_Apres(liste* prd, typelem E) ; sont supposées prédéfinies