

Epreuve Finale : Algorithmique avancée et Complexité

(1h15 min)

Exercice 1 : (10 pts)

Déterminer l'invariant des différentes boucles et en déduire l'ordre de complexité de l'algorithme suivant :

```
Recursive(int T)
Début
Si T = 1 alors retourner 1;
Sinon Retourner (Recursive (T/2));
Fsi;
Fin;

Début
Pour i=1 à N
Faire pour j=i+1 à M
    Faire pour k= i+1 à N
        Faire <.....> fait ;
    Fait;
Fait;
i:=2 tant que i<= (2N)
Faire
    si (i<=N) alors
Afficher(recursive_1(i));

    Sinon
    tant que i<=2N faire i = i*i; fait;
    Fsi;
Fait;
Fin.
```

Exercice 2 : (10 pts)

Soit en entrée un tableau d'entiers de dimension N, noté T, on cherche à extraire du tableau, deux éléments dont la somme est égale à une valeur en entrée X. Autrement dit, trouver deux éléments du tableau, T[i] et T[j] tel que $i \neq j$, $T[i] + T[j] = X$;

1. Proposer un algorithme naïf quadratique permettant de répondre au problème. **(3 pts)**
 - a. A quoi correspond le meilleur cas ? Le pire cas ? En déduire l'ordre de complexité dans chaque cas **(2 pts)**
2. Proposer un algorithme optimisé pour répondre au même problème. **(3 pts)**
 - a. A quoi correspond le meilleur cas ? Le pire cas ? En déduire l'ordre de complexité dans chaque cas **(2 pts)**