

<p>Lycée S: Remada Année scolaire: 2010/2011 Date : 06/12/2010 Durée : 1H Coef: 3</p>	<p>EPREUVE ALGORITHMIQUE <u>DEVOIR DE Synthèse N°1</u></p>	<p>Enseignante: Mlle Moufida Madhbouh Classe: 4ème SI</p>
--	---	---

Exercice 1 : (4 points)

Soit la fonction inconnue suivante :

```

0. Def fn inconnue(n :entier) :entier
1. Convch(n,ch)
2. X←long(ch)-1
3. Si n<10 alors
   Inconnue ← n
   Sinon
   Inconnue ← inconnue (n div 10) +(n mod 10)* 10^x
   Fin si
4. Fin inconnue

```

Questions :

- Exécuter manuellement cette fonction pour les appels suivants et donner les valeurs retournées:
Inconnue(25)→.....
Inconnue(214)→.....
- Quel est le rôle de cette fonction :

Exercice 2 (6points) :

Pour calculer le produit de deux entiers, on peut procéder comme suit :

$$A*B = \underbrace{A + A + A + \dots + A}_{B \text{ fois}}$$

Exemple: A=2, B=5

$$A*B = \underbrace{2 + 2 + 2 + 2 + 2}_{5 \text{ fois}}$$

- Calculer A*B tel que A=2 B=4 en utilisant le même démarche que l'exemple
- Déterminer une relation entre A*B et A*(B-1)
- Déterminer A*B si A=0 ou B=0
- Proposer une analyse puis déduisez un algorithme intitulé produit qui calcule A*B d'une manière récursive selon le principe indiqué ci-haut

Problème (12 points)

Le but de ce programme est de proposer une solution de gestion d'un ensemble d'informations concernant des élèves.

1°)

- a) Déclarer, en algorithmique, une structure enregistrement permettant de représenter un élève, sachant qu'un élève est caractérisé par :
 - ❖ Nom : 30 caractères au maximum
 - ❖ Prénom : 30 caractères au maximum
 - ❖ Lycée : 20 caractères au maximum
 - ❖ Résultat (A ou R avec A s'il est admis et R s'il est refusé)
- b) Déclarer, en algorithmique, une structure fichier permettant de stocker un ensemble des élèves.

2°) On considère que :

- « élève.dat » est un fichier des élèves (qui se trouve sur la racine c:\)
 - « Admis.txt » est un fichier texte (qui se trouve sur la racine c:\)
 - « E » est une variable de type élève
- a- Ecrire l'analyse et l'algorithme d'un module permettant de remplir le fichier « élève.dat » par n élèves ($1 \leq n \leq 15$)
 - b- Ecrire l'analyse et l'algorithme d'un module permettant d'afficher tous les noms des élèves
 - c- Ecrire l'analyse et l'algorithme d'un module permettant de stocker tous les élèves admis (dans le fichier texte « **admis.txt** »).
 - d- Ecrire l'analyse et l'algorithme d'un module permettant l'affichage du contenu de « **admis.txt** ».
 - e- Ecrire l'analyse et l'algorithme d'un module permettant l'affichage des informations d'un élève donné. On suppose que l'on connaît sa position P de l'élève dans le fichier.
 - f- Ecrire l'analyse et l'algorithme d'un module permettant la modification du résultat d'un élève donné dont on connaît sa position P dans le fichier.
 - g- Ecrivez l'algorithme du programme principal avec menu permettant les opérations suivantes :
 - Remplir le fichier « élève.dat » par n élèves
 - Afficher les noms de tous les élèves sur l'écran

- Afficher les informations des élèves admis
- Afficher les renseignements concernant un élève à partir de sa position **P** dans le fichier
- Modifier le résultat concernant un élève,

Bon travail