

» Remarque : Vous devez résoudre tous les parties de l'examen sur vos propres copies (double feuilles).

**Exercice n° 01 : (4 pts)**

Soit le programme Pascal suivant :

<pre>Program Inconnu ; Uses winct ; Var   f, g : ..... ;   nb, p, k : ..... ;   ph : String ; Begin   Assign (f, 'C:\f.txt') ;   Assign (g, 'C:\g.txt') ;   Reset (f) ;   Rewrite (g) ;   nb := 0 ;   While not eof(f) Do   Begin     Readln(f) ;     nb := nb + 1 ;   End ;   Repeat     Write ('Donner la position p :') ;     Readln (p) ;     Write ('Donner la position k :') ;     Readln (k) ;   Until (p in [1..nb]) and (k in [p..nb]) ;</pre>	<pre>Reset (f) ; nb := 0 ; While not eof (f) Do Begin   nb := nb + 1 ;   if (nb &gt;= p) And (nb &lt;= k) then     Readln (f)   Else   Begin     Readln (f, ph) ;     Writeln (g, ph) ;   End ; End ; Close (f) ; Close (g) ; Erase (f) ; Rename (g, 'C:\f.txt') ; End.</pre>
---	---

1. Compléter la partie déclarative de ce programme.

2. Donner le rôle de chaque traitement (pris du programme) présenté ci-dessous :

Traitement	Rôle
<pre>While not eof(f) Do Begin   Readln(f) ;   nb := nb + 1 ; End ;</pre>	
<pre>Erase (f) ;</pre>	
<pre>Rename (g, 'C:\f.txt') ;</pre>	



3. Soit le contenu du fichier **f.txt** avant l'exécution du programme :

```
Informatique
Réseau
Internet
Réseau sans fils
Intranet
```

⇒ Déterminer le nouveau contenu du fichier **f.txt** après l'exécution du programme (avec **p=2** et **k=4**)

4. Déduire le rôle de ce programme.

### Exercice n° 02 : (4 pts)

Soit la fonction suivante :

```
Fonction ghost (x,n :entier) :.....
  Si (n=1) alors
    ghost ← (n=1)
  Sinon si (x mod n =0) alors
    ghost ←(x < n-1)
  sinon
    ghost ←ghost (x,n-1)
  finSi
Fin ghost
```

- 1) Quelle-est le type de la fonction ?
- 2) Exécuter cette fonction pour  $x=7$  et  $n=6$ . Laisser une trace de l'exécution pas à pas.
- 3) Quelle-est le rôle de cette fonction si les paramètres d'appel sont  $x$  et  $x-1$  : **ghost** ( $x,x-1$ )

### PROBLEME : (12 pts)

Une entreprise veut informatiser la gestion de ses employés, en sauvegardant dans un fichier nommé '**Employe.dat**' ses fiches de renseignements, qui sera créé dans le dossier « société » de la partition **C**.

Chaque fiche comporte les renseignements suivants :

- Nom '**NOM**' chaîne de 10 caractères au maximum commençant par une lettre majuscule.
- Etat social '**ETAT**' : composé par les renseignements suivants
  - \* Etat civil '**CIVIL**' : "0" si l'employé est célibataire ou "1" s'il est marié.
  - \* Nombre d'enfants '**ENFANTS**' : entier non signé ('**ENFANTS**' il sera initialisé automatiquement à 0 si l'employé est célibataire)
- Salaire '**SALAIRE**' : sachant que le salaire sera attribué selon trois grades :
  - \* **Grade A** → le salaire est de 250 dinars
  - \* **Grade B** → le salaire est de 350 dinars
  - \* **Grade C** → le salaire est de 450 dinars
- Prime '**PRIME**' : dont le montant sera attribué comme suit :  
(N.B : les primes sont les mêmes pour les différents grades d'employés)
  - \* Si l'employé est **célibataire** : le prime forfaitaire est de 2.500 dinars
  - \* Si l'employé est **marié** : le montant de prime sera de :

- ✓ 4.500 dinars si l'employé n'a pas d'enfant.
- ✓ 15 dinars par enfant, si le nombre d'enfant est de 1 à 3.
- ✓ 10 dinars par enfants, si le nombre d'enfant est supérieur à 4.

Ecrire un programme nommé gestion qui permet d'assurer respectivement les actions suivantes :

- Saisir et enregistrer, au fur est à mesure, dans le fichier '**Employe.dat**' les données relatives à des employés dont le nombre n'est pas limité (l'utilisateur décide d'arrêter la saisie ou non en répondant à la question 'Voulez vous continuer (**O\N**) ? ' Après chaque saisie : Ainsi, la saisie s'arrêtera lorsque l'utilisateur répond par '**N**').
- Afficher le nom et le montant de prime **des employés mariés**.
- Insérer un employé dans une position '**P**' donnée.
- Afficher le fichier après l'insertion du nouveau employée.

**Question :**

1. Analyser ce problème en le décomposant en modules et en respectant la démarche proposée dans les énoncés.
2. Analyser les modules envisagés.
3. En déduire l'algorithme du programme principal ainsi que ceux des modules envisagés.

