

Questions de cours (06 points)

1. Donner la définition des termes suivants (une ligne au maximum pour chaque terme) : (3 points)
 - a. Boucle, Tableau, Procédure, Fonction.
2. Pourquoi on utilise les boucles dans un algorithme ? (0,5 point)
3. Citer les trois types de boucles avec ses syntaxes en langage C (1,5 points)
4. Quelle est la différence entre une **procédure** et une **fonction**? (1 point)

Exercice 1 (6 points)

Soit l'algorithme ci-contre :

- a) Que fait l'algorithme **Liste**.
- b) Que fait la fonction **estPaire()** ?
- c) Citer les variables **Globales** de l'algorithme et les variables **Locales**, si elle existes, de la fonction **estPaire()**.
- d) Citer les paramètres **Formels** et les paramètres **Effectifs** dans l'algorithme **Liste**.
- e) Donner le résultat d'exécution de l'algorithme sachant que l'utilisateur a entré la valeur **6** pour la variable **n**.
- f) Traduire l'algorithme en programme en langage C.

```

0. Algorithme Liste;
1. Var i ,n :entier;
2. Fonction estPaire(x :entier) :booléen;
3. Début
4.   Si (x mod 2=0) Alors
5.     Retourne vrai ;
6.   Sinon
7.     Retourne faux ;
8.   FinSi;
9. Fin
10. Debut
11.   écrire("donner un nombre positif") ;
12.   lire(n) ;
13.   Pour i de 1 à n Faire
14.     Si (estPaire(i)) Alors
15.       écrire(i," ") ;
16.   FinSi
17. FinPour
18. Fin
    
```

Exercice 2 (4 points)

Soit une matrice M[2,4] et deux vecteurs V1[4] et V2[4] de types entiers. Ecrire un programme en langage C qui permet de:

- a) Lire les éléments de la matrice M.
- b) Copier la première ligne de la matrice M dans le vecteur V1 et la seconde ligne de M dans le vecteur V2. Comme il est illustré dans l'exemple suivant :

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \Rightarrow V1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}; V2 = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

Exercice 3 (4 points)

- a) Dérouler l'algorithme **Vecteur** en mémoire.
- b) Donner le contenu du vecteur V[] à la fin de l'exécution.

```

0. Algorithme Vecteur;
1. Var i ,n :entier;
2.   V :tableau[1..4]d'entier ;
3. Début
4.   n ← 10 ;
5.   i ← 1 ;
6.   V[i] ← n ;
7.   Tant que (i<4) Faire
8.     i ← i+1 ;
9.     V[i] ← V[i-1]*2 ;
10. FinTantQue;
11. Fin
    
```