

Questions de cours (06 points)

Voir le support de cours.

Solution de l'exercice 1 (6 points)

a) Que fait l'algorithme **Liste ()**? (1 point)

- ✓ l'algorithme **Liste** permet d'afficher la liste des nombres pairs  $\in$  à l'intervalle  $[0..n]$

b) Que fait la fonction **estPaire()** ? (1 point)

- ✓ la fonction **estPaire(x :entier)** permette de vérifier si le nombre **x** passé en paramètres est un nombre pair ou non, si **x** est pair elle retourne **vrai** si non elle retourne **faux**.

c) Citer les variables **Globales** de l'algorithme et les variables **Locales**, si elles existent, de la fonction **estPaire()**. (1 point)

- ✓ Les variables globales sont **i, n :entier**.
- ✓ Aucune variable locale n'est déclarée dans la fonction **estPaire()**.

d) Citer les paramètres **Formels** et les paramètres **Effectifs** dans l'algorithme **Liste**. (0.5 point)

- ✓ On a un seul paramètre formel dans la fonction **estPaire()** qui est « **x :entier** », et un seul paramètre effectifs dans l'appel de la fonction **estPaire()** qui est « **i** ».

e) Donner le résultat d'exécution de l'algorithme sachant que l'utilisateur a entré la valeur **6** pour la variable **n**. (1 point)

- ✓ l'algorithme affichera les valeurs suivantes à la fin de son exécution : **2 4 6**

f) Traduire l'algorithme en programme en langage C. (1.5 points)

```
#include<stdio.h>

int i,n ;

int estPaire(int x){
if(x%2==0){
    return 1;
}else{
    return 0;
}
}

main(){
printf("donner un nombre positif");
scanf("%d",&n) ;
for(i=1 ;i<=n ;i++){
    if(estPaire(i)){
        printf("%d ",i);
    }
}
}
```

### Solution de l'exercice 2 (4 points)

```
#include<stdio.h>
int i,j,M[2][4],V1[4],V2[4];
main(){
    for(i=0;i<2;i++){
        for(j=0;j<4;j++){
            printf("donnez M[%d][%d]\n",i,j) ;
            scanf("%d",&M[i][j]);
        }
    }
    for(i=0;i<4;i++){
        V1[i]=M[0][i];
        V2[i]=M[1][i];
    }
}
```

1 point

1.5 points

1.5 points

### Solution de l'exercice 3 (4 points)

Déroulement de l'algorithme Vecteur en mémoire (3.5 points)

N°	i	n	V[1]	V[2]	V[3]	V[4]
1	?	?	∅	∅	∅	∅
2	?	?	?	?	?	?
4	?	10	?	?	?	?
5	1	10	?	?	?	?
6	1	10	10	?	?	?
8	2	10	10	?	?	?
9	2	10	10	20	?	?
8	3	10	10	20	?	?
9	3	10	10	20	40	?
8	4	10	10	20	40	?
9	4	10	10	20	40	80

Le contenu du vecteur V[] (0.5 point)

V= 

10	20	40	80
----	----	----	----