

Prénom & Nom : Classe :

✂ Les réponses à la **PARTIE I** doivent être rédigées sur cette même feuille qui doit être remise à la fin de l'épreuve avec la feuille de copie qui contiendra les réponses à la **PARTIE II**.

Partie I

(11 points)

Exercice N° 1 (3 pts)

Soit la fonction « recherche » suivante :

```
0) DEF FN recherche (T:tab; X, N:entier) : Booléen
1) i ← 0
   Répéter
   I ← i+1
   Ok ← X=T[i]
   Jusqu'à (ok) ou (i=n)
2) recherche ← ok
3) Fin recherche
```

❶ Identifier cette méthode de recherche (1pts)

.....

❷ Transformer l'instruction N° 1 en boucle « Tant Que »(1.5pts)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice N° 2 (3.5 pts)

Soient les déclarations Pascal suivantes :

```

Type
Couleurs = (Bleu, Vert, Rouge, Noir, Blanc) ;
Tab = Array [Bleu. . Blanc] of Char ;

Var
V: tab;   r: real; i: couleurs ; f:integer;

Procedure P1 (var x: integer);
  Var n,x1: integer;
  Function Somme (m:integer):integer;
    var q,y:integer;
    Begin
      q:= m DIV 3;
      y:= m MOD 3;
      Somme:= q * y ;
    End;

begin
  readln (n);
  x: =Random (n);
  s:= x + Somme(x);
end;

```

Compléter le tableau ci-dessous, en mettant dans la 2^{ème} colonne la lettre **V** si la proposition donnée dans la 1^{ère} colonne est valide ou par **F** dans le cas contraire, tout en justifiant la réponse si l'instruction est non valide.

Proposition	Validité de la proposition	Justification (Si Faux)
For i := Vert To Blanc Do V[i] := Chr (ORD(i) +65) ;		
R := 15.23 ; Writeln(R : 8 :2) ; Resultat affiché est 15.23000		
Pour appeler la fonction Somme dans le programme principale, on utilise cette instruction : f :=Somme(20) ;		
Les objets q, y <u>sont reconnus</u> par la procédure P1		
Une fonction <u>peut être toujours</u> transformée en une procédure		
For i := Bleu To Blanc Do Write(i) ;		

Exercice N° 3 (4.5 pts)

Soit l'algorithme de la fonction suivante :

```
function inconnue(n:integer):.....;
var
..... ;
begin
d:=2;
while (n mod d <> 0) and (n div 2 >= d) do
begin
d:=d+1;
end;
inconnue:= n mod d <> 0 ;
end;
```

- ❶ Compléter les pointillés ? (1pts)
- ❷ Déterminer inconnue(9) puis inconnue(11) ? (1pts)

.....

.....

- ❸ En déduire le rôle de cette fonction ? (0.5pts)

.....

.....

- ❹ Un **Nombre premier N** est dit **circulaire** s'il vérifie la propriété suivante : **chacune des rotations de ses chiffres d'un élément vers la droite, forme à son tour un nombre premier.**

Exemples pour :

- N=719, N est nombre premier circulaire car 719, 971 et 197 sont des nombres premiers
- N=23, N n'est pas un nombre premier circulaire car il est premier mais 32 ne l'est pas.
- N=6102, N n'est pas un nombre premier circulaire car il n'est pas premier

Écrire l'algorithme d'une fonction qui permet de vérifier pour un entier N donné s'il est **premier circulaire ou non** (2pts)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ecrire un programme qui permet de saisir une chaîne CH alphabétique majuscule et d'afficher si cette chaîne **sommet palindrome** ou **non**

Une chaîne est dite palindrome lorsqu'on peut lire dans le deux sens.

Une chaîne est dite sommet palindrome lorsqu'elle est :

- Palindrome
- Le premier, le dernier et le caractère du milieu sont identiques.

Pour cela on doit suivre les étapes suivantes :

- Saisir un entier N, qui représente la longueur d'une chaîne CH sachant que N est entier premier compris entre 3 et 9.
- Saisir le premier et le dernier caractère de la chaîne CH.
- Le reste des caractères sont saisis d'une façon aléatoire.
- Vérifier si cette chaîne **sommet palindrome** ou **non**.

N.B :

✎ un entier est dit premier si et seulement si divisible par 1 et lui-même.

✎ ORD ('A')=65

Exemple :

N=7(entier premier) et ch="ALLALLA" le programme affichera **sommet palindrome**

N=5 (entier premier) et ch= "RADAR" le programme affichera **palindrome non sommet**

N=3 et ch="AMI" le programme affichera **ni sommet palindrome ni palindrome**

Travail demandé

- ❶ Analyser le problème en le décomposant en modules.
- ❷ Analyser chaque module proposé.