

Nom : **OUARET**

Prénom : **Ahmed**

Groupe : F1

Exercice :

Soit le programme Pascal suivant :

```

program perimetreR;
var long, larg, P : integer;
Procedure PerimetreRectangle(longueur, largeur: integer ;
                             var Perimetre : integer);
begin
    Perimetre := 2 * (longueur + largeur);
end;

BEGIN
    read(long, larg);
    PerimetreRectangle(long, larg, P);
    write('Le perimetre du rectangle est :', P);
END.
    
```

Questions :

1. Identifier le type de sous-programme présent dans ce programme.
2. Complétez le programme avec ce qui manque.
3. Dérouler le programme pour long=7 et larg=5.
4. Déduire ce que fais le programme.
5. Réécrire le programme en transformant le sous-programme utilisé à l'autre type de sous-programme.

Réponse :

1/ Le type de sous-programme présent dans ce programme est : **procédure**

2/ Complétez le programme avec ce qui manque : **procedure, integer** (voir le programme)

3/ Dérouler le programme pour long=7 et larg=5.

Instruction	Prog. Principal			La procédure PerimetreRectangle			Affichage
	long	larg	P	longueur	largeur	Perimetre	
read(long, larg);	7	5					
PerimetreRectangle(long, larg, P); // l'appel à la procédure PerimetreRectangle avec long=7 et larg=5							
// Transmission des paramètres Perimetre :=2*(longueur + largeur);				7	5	24	
// Retour du résultat dans le programme principale PerimetreRectangle(long, larg, P);		24					
write('Le perimetre du rectangle est :', P);							Le perimetre du rectangle est : 24

4/ Déduire ce que fais le programme.

Le programme "perimetreR" a pour but de calculer le périmètre d'un rectangle à partir de sa longueur et de sa largeur.

5/ Réécrire le programme en transformant le sous-programme utilisé à l'autre type de sous-programme.

```
1  program perimetreR;
2  var long, larg, P : integer;
3  Function PerimetreRectangle(longueur, largeur: integer):integer ; // mot clé function & le type de la fonction
4  var Perimetre:integer; // nouvelle variable : var Perimetre:integer;
5  begin
6  Perimetre := 2 * (longueur + largeur);
7  PerimetreRectangle:=Perimetre; // l'instruction : PerimetreRectangle:=Perimetre;
8  end;
9
10 BEGIN
11 read(long, larg);
12 P:=PerimetreRectangle(long, larg); // l'appel à la fonction
13 write('Le perimetre du rectangle est :', P);
14 END.
```

Nom : **OUARET**

Prénom : **Ahmed**

Groupe : F1

Exercice :

Soit le programme Pascal suivant :

```

program perimetreR;
var longueur, largeur, Perimetre : integer;
Procedure PerimetreRectangle(long, larg: integer ;
                        var P : integer);

begin
    P := 2 * (long + larg);
end;

BEGIN
    write('Introduire la longueur et la largeur du rectangle');
    read(longueur, largeur);
    PerimetreRectangle(longueur, largeur, Perimetre);
    write('Le perimetre du rectangle est :', Perimetre);
END.
    
```

Questions :

1. Identifier le type de sous-programme présent dans ce programme.
2. Complétez le programme avec ce qui manque.
3. Dérouler le programme pour longueur=4 et largeur=3.
4. Déduire ce que fais le programme.
5. Réécrire le programme en transformant le sous-programme utilisé à l'autre type de sous-programme.

Réponse :

1/ Le type de sous-programme présent dans ce programme est : **procédure**

2/ Complétez le programme avec ce qui manque : **procedure, integer** (voir le programme)

3/ Dérouler le programme pour longueur=4 et largeur=3.

Instruction	Prog. Principal			La procédure PerimetreRectangle			Affichage
	longueur	largeur	Perimetre	long	larg	P	
write('Introduire la longueur et la largeur du rectangle');							Introduire la longueur et la largeur du rectangle
read(long, larg);	4	3					
PerimetreRectangle(long, larg, P); // l'appel à la procédure PerimetreRectangle avec long=7 et larg=5							
// Transmission des paramètres P :=2*(long + larg);				4	3	14	
// Retour du résultat dans le programme principale PerimetreRectangle(longueur, largeur, Perimetre);		14					
write('Le perimetre du rectangle est :', P);							Le perimetre du rectangle est : 14

4/ D  duire ce que fait le programme.

Le programme "PerimetreRectangle" a pour but de calculer le p  rim  tre d'un rectangle    partir de sa longueur et de sa largeur.

5/ R   crire le programme en transformant le sous-programme utilis      l'autre type de sous-programme.

```
1  program perimetreR;
2  var longueur, largeur, Perimetre : integer;
3  Function PerimetreRectangle(long, larg: integer):integer ; // mot cl   fonction & le type de la fonction
4  var P:integer;      // nouvelle variable      : var P:integer;
5  begin
6  P := 2 * (long + larg);
7  PerimetreRectangle:=P; // l'instruction : PerimetreRectangle:=P;
8  end;
9
10 BEGIN
11 read(longueur, largeur);
12 Perimetre:=PerimetreRectangle(longueur, largeur); // l'appel    la fonction
13 write('Le perimetre du rectangle est :', Perimetre);
14 END.
15
```