

Corrigé de l'Examen Informatique-1

L'usage de la calculatrice est interdit

Durée : 01h30 min

Exercice 01 : [07 Points]

1. Citer deux particularités de la boucle **Répéter (Repeat)** ? 1pts

- 0.5 x 2 Pts** {
- ✓ **La première itération est toujours exécutée quelque soit la condition de sortie de la boucle.**
 - ✓ **La condition de sortie est vérifiée après l'exécution du bloc d'un programme.**
 - ✓ **Permet de répéter un bloc d'instructions jusqu'à ce qu'une condition spécifique soit remplie.**
 - ✓ **Contrôle explicite de la condition d'arrêt.**
 - ✓ **La boucle Répéter est particulièrement adaptée utilisable si l'on ne connaît pas la condition d'arrêt exacte dès le début.**

2. Quel est le rôle de la compilation d'un programme? 1pts

- 0.50 {
- ✓ **Vérification de la syntaxe** : La compilation permet de détecter et signaler les erreurs de syntaxe dans le code, assurant ainsi que le programme est correctement écrit avant d'être exécuté.
 - ✓ **Traduction du code source** : La compilation transforme le code source écrit par le programmeur en un langage machine que l'ordinateur peut comprendre et exécuter.
 - ✓ **Création d'un fichier exécutable** : Elle génère un fichier exécutable qui permet à l'ordinateur de lancer le programme sans avoir besoin du code source.

3. Citer cinq types de variables couramment utilisés en programmation Pascal.? 1pts

0.125	0.125	0.25	0.25	0.25
Entier (Integer)	Flottant (Float)	Caractère (Character)	Chaîne de caractères (String)	Booléen (Boolean)

4. Effectuer la conversion suivante : $(402)_8 = (?)_{16}$ 1pts

Pour convertir un N^{br} de la base 8 à la base 16, il faut passer par une base intermédiaire (Base 2 ou Base 10) :

Méthode 1 : Passage par la base 2 :					
En utilisant la table :	Hexadécimal (Base 16)	Chiffres équivalents en Binaire			
		2^3	2^2	2^1	2^0
	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	0	1	1
	4	0	1	0	0
	5	0	1	0	1
	6	0	1	1	0
	7	0	1	1	1
	8	1	0	0	0
	9	1	0	0	1
	A	1	0	1	0
	B	1	0	1	1
	C	1	1	0	0
	D	1	1	0	1
	E	1	1	1	0
	F	1	1	1	1

Base8

0.50 {

Donc : $(402)_8 = (102)_{16}$

0.50 {

➤ **Groupe de 3 bits $2^3=8$**
 $(4)_8 = (100)_2$;
 $(0)_8 = (000)_2$;
 $(2)_8 = (010)_2$;

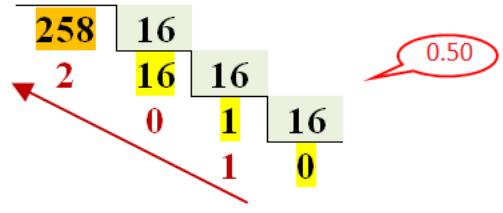
➤ **Groupe de 4 bits $2^4=16$**
 $(0010)_2 = (2)_{16}$;
 $(0000)_2 = (0)_{16}$;
 $(0001)_2 = (1)_{16}$;

Méthode 2 : Passage par la base 10 :

$$(402)_8 = (2 \cdot 8^0 + 0 \cdot 8^1 + 4 \cdot 8^2)_{10}$$

$$= (2 + 0 + 256)_{10}$$

$$= (258)_{10}$$



$$(402)_8 = (102)_{16}$$

5. Trouver les indices de base x et y des conversions suivantes: $(102)_x = (18)_{10}$ et $(25)_y = (37)_{10}$ Ipt

► $(102)_x = (18)_{10}$

$$(102)_x = 2 \cdot x^0 + 0 \cdot x^1 + 1 \cdot x^2$$

$$= 2 + x^2$$

$$2 + x^2 = 18 \rightarrow x^2 = 16 \rightarrow x = \sqrt{16} \rightarrow x = 4$$

► $(25)_y = (37)_{10}$

$$(25)_y = 5 \cdot y^0 + 2 \cdot y^1$$

$$= 5 + 2y$$

$$5 + 2y = 37 \rightarrow 2y = 37 - 5 \rightarrow y = 32 / 2 \rightarrow y = 16$$

6. Evaluer les expressions suivantes en montrant l'ordre des opérations : 2pts

► $E_I = (x > y) OR (z >= w) AND (x < > w) OR NOT k;$ pour: $x=7, y=5, z=3, w=1$ et $k=True.$

$$E_I = (7 > 5) OR (3 >= 1) AND (7 < > 1) OR NOT True; \quad (7 > 5) \rightarrow True$$

$$E_I = True OR (3 >= 1) AND (7 < > 1) OR NOT True; \quad (3 >= 1) \rightarrow True$$

$$E_I = True OR True AND (7 < > 1) OR NOT True; \quad (7 < > 1) \rightarrow True$$

$$E_I = True OR True AND True OR NOT True; \quad NOT True \rightarrow False$$

$$E_I = True OR True AND True OR False; \quad True AND True \rightarrow True$$

$$E_I = True OR True OR False; \quad True OR True \rightarrow True \text{ ou bien } True OR False \rightarrow True$$

$$E_I = True OR False; \quad True OR False \rightarrow True$$

$$E_I = True; \quad True$$

► $E_2 = \frac{c + 4\sqrt{d}}{b-a}; \text{ avec : } a=1, b=4, c=3 \text{ et } d=9$

$$E_2 = (c + \sqrt{d}) * 4 / (b-a)$$

$$E_2 = (3 + \sqrt{9}) * 4 / (4-1)$$

$$E_2 = (3 + 3 * 4) / (4-1)$$

$$E_2 = (3 + 12) / (4-1)$$

$$E_2 = 15 / (4-1)$$

$$E_2 = 15 / 3$$

$$E_2 = 5$$

Exercice 02 : [08Points]

1. Traduire l'algorithme en un programme PASCAL : 2.5pts

Algorithme	Programme en Pascal
<p>Algorithme Exercice02; Variables N, i, M, T : entier; Début Lire (N); $M \leftarrow 0$; $T \leftarrow 1$; Pour $i \leftarrow 1$ à N faire Si ($i \bmod 2 = 0$) alors $M \leftarrow M + (i * 3)$; Sinon $M \leftarrow M + (i * i)$; $T \leftarrow T * (i + 1)$; FinSi ; FinPour ; Écrire ('$M =$', M, ' $T =$', T) ; Fin .</p>	<pre> program Exercice02; 0.125 var 0.125 N, i, M, T: integer; 0.125 begin 0.125 Read(N); 0.125 M := 0; T := 1; 0.25 for i := 1 to N do 0.125 begin 0.125 if (i mod 2 = 0) then 0.125 M := M + (i * 3) 0.25 else 0.125 begin 0.125 M := M + (i * i); 0.125 T := T * (i + 1); 0.125 end; 0.125 end; 0.125 Write('M = ', M, ' T = ', T); 0.125 end. 0.125 </pre>

2. Dérouler l'algorithme pour $N=3$. 1.5pts

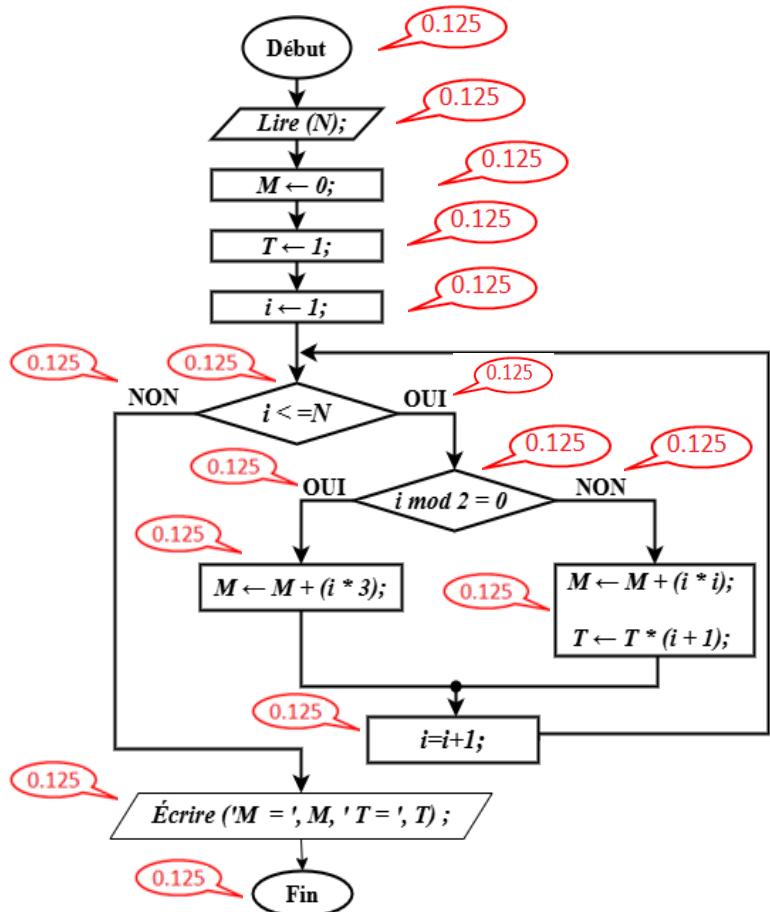
Instructions	Variables				Affichage
	N	i	M	T	
Lire (N);	3 0.125	/	/	/	/
$M \leftarrow 0$; $T \leftarrow 1$;	3	/	0 0.125	1	
Pour $i \leftarrow 1$ (<i>On exécute le bloc SINON</i>) $M \leftarrow M + (i * i)$; $M \leftarrow 0 + (1 * 1)$; $M \leftarrow 1$; $T \leftarrow T * (i + 1)$; $T \leftarrow 1 * (1 + 1)$; $T \leftarrow 2$;	3	1 0.125	1 0.125	2 0.125	/
Pour $i \leftarrow 2$ (<i>On exécute le bloc SI</i>) $M \leftarrow M + (i * 3)$; $M \leftarrow 1 + (2 * 3)$; $M \leftarrow 7$;	3	2 0.125	7 0.125	2 0.125	/
Pour $i \leftarrow 3$ (<i>On exécute le bloc SINON</i>) $M \leftarrow M + (i * i)$; $M \leftarrow 7 + (3 * 3)$; $M \leftarrow 16$; $T \leftarrow T * (i + 1)$; $T \leftarrow 2 * (3 + 3)$; $T \leftarrow 8$;	3	3 0.125	16 0.125	8 0.125	/
Écrire (' $M = $ ', M , ' $T = $ ', T);	3	3	16	8	$M = 16 \ T = 8$ 0.125

3. L'expressions générales de M et T sont : 1pts

$$M = \sum_{\substack{i=1 \\ i_paire}}^N (i * 3) + \sum_{\substack{i=1 \\ i_impaire}}^N i^2$$

$$T = \prod_{\substack{i=1 \\ i_impaire}}^N (i + 1)$$

4. L'organigramme de l'algorithme est : 2pts



5. Réécrire le programme PASCAL en remplaçant la boucle For par la boucle repeat : 1pts

```

program Exercice02_repeat;
var
    N, i, M, T: integer;
begin
    Read(N);
    M := 0;
    T := 1;
    i := 1; repeat
        if (i mod 2 = 0) then
            M := M + (i * 3)
        else
            begin
                M := M + (i * i);
                T := T * (i + 1);
            end;
        i := i + 1; until i > N;
    Write('M = ', M, ' T = ', T);
end.

```

Exercice 03 : [05 Points]

Ecrire un programme en **PASCAL** qui demande à l'utilisateur d'introduire une heure $h \in [0, 24]$ de son choix et des minutes $m \in [0, 59]$. Le programme doit ensuite afficher l'heure correspondante en ajoutant une minute en prenant en compte le changement d'heure si nécessaire.

Exemple : Si on introduit 15h30min, le programme affichera 15h31min.

Programme en PASCAL
<pre>program Exercice03; { 0.25 var { 0.125 h, m: integer; { 0.25 begin { 0.125 writeln('Veuillez introduire h et m'); { cette instruction est facultative } readln(h,m); { 0.25 if (h < 0) or (h >= 24) then { 0.25 WriteLn('Lheure doit être comprise entre 0 et 23.') { 0.25 else if (m < 0) or (m >= 60) then { 0.25 WriteLn('Les minutes doivent être comprises entre 0 et 59.') { 0.25 else { 0.25 begin { 0.25 m := m + 1; { 0.25 if m = 60 then { 0.25 begin { 0.25 m := 0; { 0.25 h := h + 1; { 0.25 end; { 0.25 if h = 24 then { 0.25 h := 0; { 0.25 { Affichage de l'heure après l'ajout d'une minute } WriteLn('Heure avec une minute de plus est : ', h, ':', m); { 0.25 end; { 0.125 end. { 0.125 end.</pre>