

## 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.

0.50

- ✓ **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 :

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

$$(402)_8 = (100\ 000\ 010)_2$$

$$(0001\ 0000\ 0010)_2 = (102)_{16}$$

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

➤ Groupe de 3 bits  $2^3=8$

$$(4)_8 = (100)_2;$$

$$(0)_8 = (000)_2;$$

$$(2)_8 = (010)_2;$$

➤ Groupe de 4 bits  $2^4=16$

$$(0010)_2 = (2)_{16};$$

$$(0000)_2 = (0)_{16};$$

$$(0001)_2 = (1)_{16};$$

Hexadécimal (Base 16)

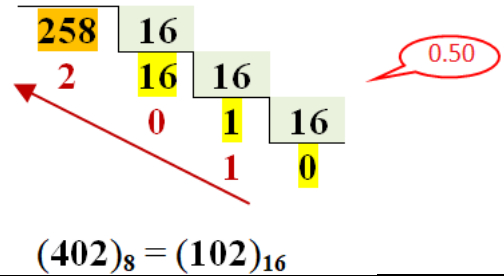
Chiffres équivalents en Binaire

	$2^3$	$2^2$	$2^1$	$2^0$
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

### Méthode 2 : Passage par la base 10 :

$$\begin{aligned}(402)_8 &= (2 \cdot 8^0 + 0 \cdot 8^1 + 4 \cdot 8^2)_{10} \\ &= (2 + 0 + 256)_{10} \\ &= (258)_{10}\end{aligned}$$



5. Trouver les indices de base x et y des conversions suivantes:  $(102)_x = (18)_{10}$  et  $(25)_y = (37)_{10}$  1pt

➤ $(102)_x = (18)_{10}$	➤ $(25)_y = (37)_{10}$
$\begin{aligned}(102)_x &= 2 \cdot x^0 + 0 \cdot x^1 + 1 \cdot x^2 \\ &= 2 + x^2\end{aligned}$ $2 + x^2 = 18 \rightarrow x^2 = 16 \rightarrow x = \sqrt{16} \rightarrow x=4$	$\begin{aligned}(25)_y &= 5 \cdot y^0 + 2 \cdot y^1 \\ &= 5 + 2y\end{aligned}$ $5 + 2y = 37 \rightarrow 2y = 37 - 5 \rightarrow y = 32 / 2 \rightarrow y=16$

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

➤  $E_1 = (x > y) \text{ OR } (z \geq w) \text{ AND } (x < w) \text{ OR NOT } k$ ; pour:  $x=7, y=5, z=3, w=1$  et  $k=\text{True}$ .

$$E_1 = (7 > 5) \text{ OR } (3 \geq 1) \text{ AND } (7 < 1) \text{ OR NOT True}; \quad (7 > 5) \rightarrow \text{True}$$

$$E_1 = \text{True OR } (3 \geq 1) \text{ AND } (7 < 1) \text{ OR NOT True}; \quad (3 \geq 1) \rightarrow \text{True}$$

$$E_1 = \text{True OR True AND } (7 < 1) \text{ OR NOT True}; \quad (7 < 1) \rightarrow \text{True}$$

$$E_1 = \text{True OR True AND True OR NOT True}; \quad \text{NOT True} \rightarrow \text{False}$$

$$E_1 = \text{True OR True AND True OR False}; \quad \text{True AND True} \rightarrow \text{True}$$

$$E_1 = \text{True OR True OR False}; \quad \text{True OR True} \rightarrow \text{True} \text{ ou bien } \text{True OR False} \rightarrow \text{True}$$

$$E_1 = \text{True OR False}; \quad \text{True OR False} \rightarrow \text{True}$$

$$E_1 = \text{True};$$

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

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

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

$$E_2 = (3 + 3 \cdot 4) / (4 - 1)$$

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

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

$$E_2 = 15 / 3$$

$$E_2 = 5$$

## Exercice 02 : [08Points]

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

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

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

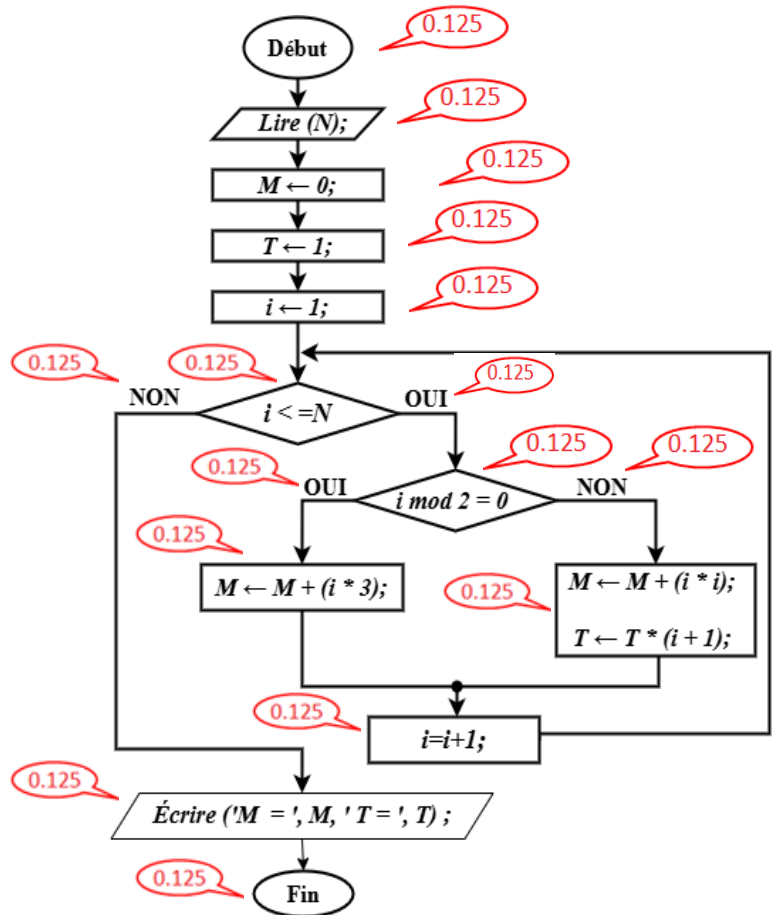
Instructions	Variables				Affichage
	N	i	M	T	
Lire (N) ;	3 0.125	/	/	/	/
$M \leftarrow 0; T \leftarrow 1;$	3	/	0 0.125	1 0.125	
<b>Pour</b> $i \leftarrow 1$ (On exécute le bloc SINON) $M \leftarrow M + (i * i); M \leftarrow 0 + (1 * 1); M \leftarrow 1;$ $T \leftarrow T * (i + 1); T \leftarrow 1 * (1 + 1); T \leftarrow 2;$	3	1 0.125	1 0.125	2 0.125	/
<b>Pour</b> $i \leftarrow 2$ (On exécute le bloc SI) $M \leftarrow M + (i * 3); M \leftarrow 1 + (2 * 3); M \leftarrow 7;$	3	2 0.125	7 0.125	2 0.125	/
<b>Pour</b> $i \leftarrow 3$ (On exécute le bloc SINON) $M \leftarrow M + (i * i); M \leftarrow 7 + (3 * 3); M \leftarrow 16;$ $T \leftarrow T * (i + 1); T \leftarrow 2 * (3 + 3); T \leftarrow 8;$	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;</pre>	0.25
<pre>var</pre>	0.125
<pre>  h, m: integer;</pre>	0.25
<pre>begin</pre>	0.125
<pre>  writeln('Veuillez introduire h et m'); { cette instruction est facultative }</pre>	
<pre>  readln(h,m);</pre>	0.25
<pre>  if (h &lt; 0) or (h &gt;= 24) then</pre>	0.25
<pre>    WriteLn('Lheure doit être comprise entre 0 et 23.')</pre>	0.25
<pre>  else if (m &lt; 0) or (m &gt;= 60) then</pre>	0.25
<pre>    WriteLn('Les minutes doivent être comprises entre 0 et 59.')</pre>	0.25
<pre>  else</pre>	0.25
<pre>    begin</pre>	0.25
<pre>      m := m + 1;</pre>	0.25
<pre>      if m = 60 then</pre>	0.25
<pre>        begin</pre>	0.25
<pre>          m := 0;</pre>	0.25
<pre>          h := h + 1;</pre>	0.25
<pre>        end;</pre>	0.25
<pre>      if h = 24 then</pre>	0.25
<pre>        h := 0;</pre>	0.25
<pre>      { Affichage de l'heure après l'ajout d'une minute }</pre>	
<pre>      WriteLn('Heure avec une minute de plus est : ', h, ':', m);</pre>	0.25
<pre>    end;</pre>	0.125
<pre>end.</pre>	0.125