

Examen de rattrapage – Informatique 1 (Corrigé Type)

Exercice 01 : [8 points]

1. Quelles sont les trois structures itératives de base ? (0.75 pts)

Les trois structures itératives de base en Pascal sont :

- La boucle Pour « for » ; 0.25
- La boucle Tant-que « While » ; 0.25
- La boucle Répéter « repeat » . 0.25

2. Citer deux périphériques d'entrée et deux périphériques de sortie. (1 pt)

- Périphériques d'entrée : Clavier, Souris. 0.25 0.25
- Périphériques de sortie : Écran, Imprimante. 0.25 0.25

Autres réponses :

Périphériques d'entrée : Scanner, Microphone, Webcam, Tablette graphique, Joystick, Pavé tactile, Périphériques de capture de mouvement, Lecteur de code-barres.

Périphériques de sortie : Haut-parleurs, Projecteur, Casque audio, Écran tactile.

3. Quels sont les identificateurs valides et invalides parmi les exemples suivants (les identificateurs non valides doivent être justifiés) : _EX01 ; EX_01 ; EX-01 ; EX01_ . (1.25 pts)

- _EX01 : Valide 0.25
- EX_01 : Valide 0.25
- EX-01 : Invalide (L'identificateur contient un caractère spécial (tiret de 6 « - »)) 0.25
- EX01_ : Valide 0.25

4. Evaluer l'expression suivante tout en montrant l'ordre des opérations :

$(a > 9 \text{ DIV } 4) \text{ AND } (a < > b) \text{ OR NOT } (c = b)$. avec $a = 4, b = 8, c = 8$. (1 pt)

$(4 > 9 \text{ DIV } 4) \text{ AND } (4 < > 8) \text{ OR NOT } (8 = 8)$. 0.125

$(4 > 2) \text{ AND } (4 < > 8) \text{ OR NOT } (8 = 8)$. 0.125

True AND $(4 < > 8) \text{ OR NOT } (8 = 8)$. 0.125

True AND True OR NOT $(8 = 8)$. 0.125

True AND True OR NOT True. 0.125

True AND True OR False. 0.125

True OR False. 0.125

True 0.125

5. Donner l'expression arithmétique correspondante à l'expression suivante écrite en Pascal :

$$\text{sqrt}(\text{abs}(2*x + 1 + x*y)) / (\text{sqr}(x) - 2*x*y) + \text{sqrt}(4*x + 5*x). \text{ (1 pt)}$$

$$\frac{\sqrt{|2x + 1 + xy|}}{x^2 - 2xy} + \sqrt{4x + 5x} \quad \text{1}$$

6. Effectuer les conversions suivantes : $(1EA)_{16} = (?)_8$, $(240)_{10} = (?)_2$. (3 pts)

- $(1EA)_{16} = (?)_8$:

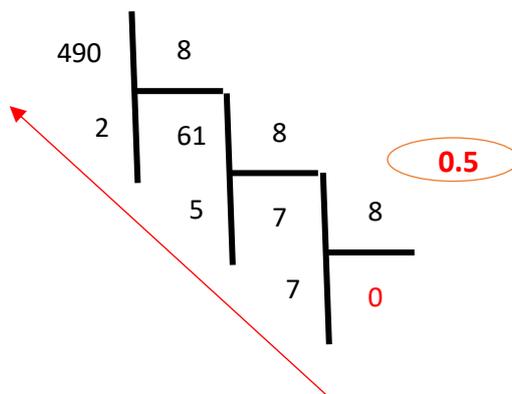
Méthode 1 :

$$(1EA)_{16} = (?)_{10} = 1 \times 16^2 + E \times 16^1 + A \times 16^0 = 1 \times 16^2 + 14 \times 16^1 + 10 \times 16^0$$

$$= (490)_{10} \quad \text{0.5}$$

$$\text{0.5}$$

$$(490)_{10} = (?)_8$$



$$(490)_{10} = (752)_8 \quad \text{0.5}$$

Méthode 2 :

$$(1EA)_{16} = (?)_2 \quad \text{0.5}$$

$$\text{0.5}$$

$$(1EA)_{16} = (0001\ 1110\ 1010)_2$$

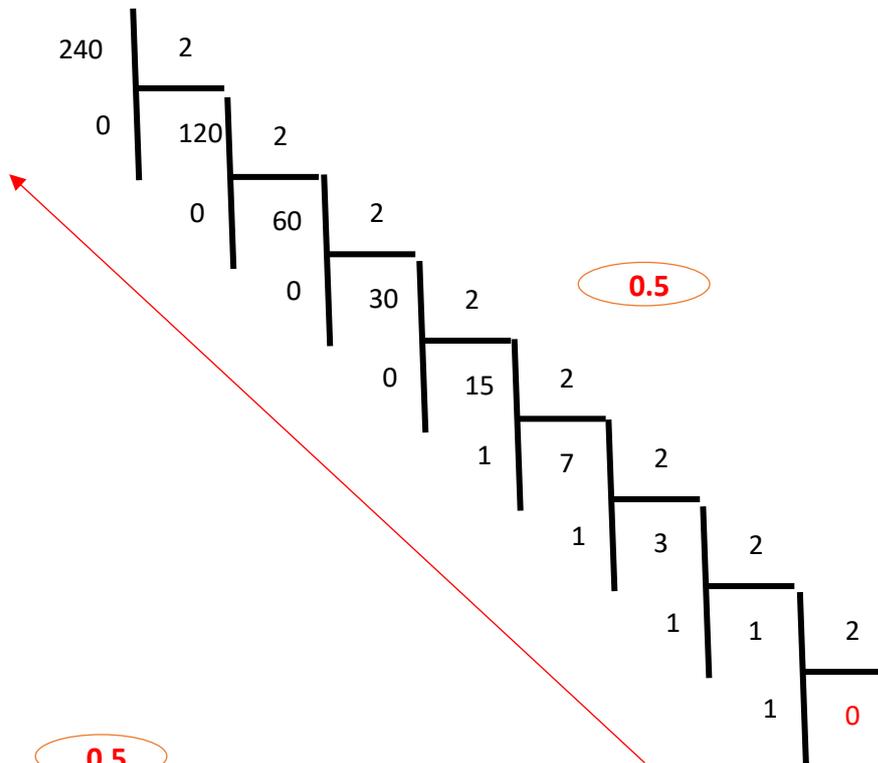
$$(000\ 111\ 101\ 010)_2 = (0752)_8$$

$$\text{0.5} \quad = (752)_8$$

$$\text{0.5}$$

Chiffre en base 16	Chiffre équivalent en base 2			
	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

• $(240)_{10} = (?)_2$



$(240)_{10} = (11110000)_2$

Exercice 02 : [8 points]

1. Traduire l'algorithme en programme PASCAL.

Algorithme	Programme PASCAL
<p>Algorithme Exo2 ;</p> <p>Variabes</p> <p>N, Sn, i : entier;</p> <p>Début</p> <p> Lire(N) ;</p> <p> Sn ← 0 ;</p> <p> Pour i ← 1 à N faire</p> <p> Si (i mod 2 = 1) alors</p> <p> Sn ← Sn + i ;</p> <p> FinSi ;</p> <p> FinPour ;</p> <p> Écrire('La somme est : ', Sn);</p> <p>Fin.</p>	<p>Program Exo2 ;</p> <p>Var</p> <p>N, Sn, i : integer;</p> <p>Begin</p> <p> read(N) ;</p> <p> Sn := 0 ;</p> <p> for i := 1 to N do</p> <p> begin</p> <p> if (i mod 2 = 1) then</p> <p> Sn := Sn + i ;</p> <p> end ;</p> <p> write('La somme est : ', Sn);</p> <p>end.</p>

2. Dérouler l'algorithme pour N = 5.

Instructions	Variables			Affichage
	i	N	Sn	
Lire(N) ;		5		0.25
Sn ← 0 ;		5	0	0.25
Pour i = 1 Si (i mod 2 = 1) VRAI Sn ← Sn + i ;	1	5	1	0.25
Pour i = 2 Si (i mod 2 = 1) FAUX	2	5	1	0.25
Pour i = 3 Si (i mod 2 = 1) VRAI Sn ← Sn + i ;	3	5	4	0.25
Pour i = 4 Si (i mod 2 = 1) FAUX	4	5	4	0.25
Pour i = 5 Si (i mod 2 = 1) VRAI Sn ← Sn + i ;	5	5	9	0.5
Écrire('La somme est : ', Sn);	5	5	9	La somme est : 9 0.5

3. Dédire ce que fait l'algorithme.

L'algorithme calcule la somme des nombres impairs compris entre 1 et N.

1

4. Réécrire le programme en remplaçant la boucle For par la boucle While.

Programme PASCAL (avec la boucle While)

```
Program Exo2 ;  
Var  
N, Sn, i : integer;  
begin  
  read(N) ;  
  Sn := 0 ; i := 1; 0.25  
  While (i <= N) do 0.5  
  begin  
    if (i mod 2 = 1) then  
      Sn := Sn + i ;  
      i := i + 1; 0.25  
    end ;  
    write('La somme est : ', Sn);  
  end.
```

5. Modifier l'algorithme pour calculer et afficher la somme « Sn » des nombres qui ne sont pas des diviseurs de N.

Programme

```
Program Exo2_5 ;  
Var  
N, Sn, i : integer;  
begin  
  read(N) ;  
  Sn := 0 ;  
  for i := 1 to N do  
  begin  
    if (N mod i <> 0) then 1  
      Sn := Sn + i ;  
    end ;  
    write('La somme est : ', Sn);  
  end.
```

Exercice 03 : [4 points]**Solution**

```
program CalculCoutStationnement; 0.25
var
heures, cout : Integer; 0.25
begin 0.25
  write('Entrez le nombre d"heures de stationnement : '); 0.25
  read(heures); 0.25
  if (heures >= 10) then 0.5
    cout := 500 0.25
  else 0.25
    if (heures > 5) then 0.5
      cout := (5 * 20) + ((heures - 5) * 50) 0.25
    else 0.25
      cout := heures * 20; 0.5
  writeln('Le coût du stationnement est de ', cout, ' DA. '); 0.25
end.
```