

Corrigé de l'examen – Informatique 1

Corrigé de l'exercice 01 : [8 points]

1. Donner les quatre éléments essentiels qui composent l'unité centrale (C.P.U). **(1 point)**

Les quatre éléments essentiels qui composent l'unité centrale (C.P.U) sont :

- Unité Arithmétique et Logique (UAL) 0.25 Pt
- Unité de Contrôle (UC) 0.25 Pt
- Mémoire vive **RAM** (Random Access Memory) 0.25 Pt
- Mémoire morte **ROM** (Read Only Memory) 0.25 Pt

2. À quoi sert l'identificateur ? **(1point)**

L'identificateur sert pour nommer un **programme**, une **variable**, une **constante**, une **fonction** ou une **procédure**

1 Pt

3. Effectuer les conversions suivantes : **(1.5 point)**

$$(120)_8 = (?)_{16} = (?)_{10}, (82)_{10} = (?)_8$$

$$(120)_8 = (001\ 010\ 000)_2 = (0000\ 0101\ 0000)_2 = (050)_{16} \text{ ou } (50)_{16}$$

0.25 Pt

0.25 Pt

$$(120)_8 = 0 \times 8^0 + 2 \times 8^1 + 1 \times 8^2 = 0 + 16 + 64 = (80)_{10}$$

0.25 Pt

0.25 Pt

2ème méthode :

Le passage par la base 10 est aussi accepté

$$(120)_8 = 0 \times 8^0 + 2 \times 8^1 + 1 \times 8^2 = 0 + 16 + 64 = (80)_{10}$$

0.25 Pt

0.25 Pt

Le poids le plus faible

80	16	
0	5	16
	5	0

Le poids le plus fort

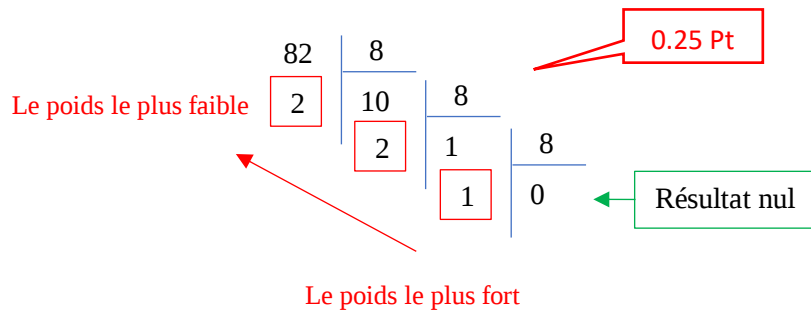
Résultat nul

0.25 Pt

$$(120)_8 = (50)_{16}$$

0.25 Pt

$$(82)_{10} = (?)_8$$



$$(82)_{10} = (122)_8$$

0.25 Pt

4. Evaluer les expressions suivantes tout en montrant l'ordre des opérations : (2.5 points)

$$E1 = (x <> y) \text{ AND NOT}(7 \text{ MOD } 3 < xy) \text{ OR } (2x > z) ; \text{ avec } x = 3, y = 4, z = 2$$

$$E1 = (3 <> 4) \text{ AND NOT}(7 \text{ MOD } 3 < 3*4) \text{ OR } (2*3 > 2)$$

1

0.125 Pt

$$E1 = \text{True AND NOT}(7 \text{ MOD } 3 < 3*4) \text{ OR } (2*3 > 2)$$

2

0.125 Pt

$$E1 = \text{True AND NOT}(1 < 3*4) \text{ OR } (2*3 > 2)$$

3

0.125 Pt

$$E1 = \text{True AND NOT}(1 < 12) \text{ OR } (2*3 > 2)$$

4

0.125 Pt

$$E1 = \text{True AND NOT True OR } (2*3 > 2)$$

5

0.125 Pt

$$E1 = \text{True AND NOT True OR } (6 > 2)$$

6

0.125 Pt

$$E1 = \text{True AND NOT True OR True}$$

7

0.125 Pt

$$E1 = \text{True AND False OR True}$$

8

0.125 Pt

$$E1 = \text{False OR True}$$

9

0.125 Pt

$$E1 = \text{True}$$

0.125 Pt

2ème méthode :

$$E1 = (3 <> 4) \text{ AND NOT}(7 \text{ MOD } 3 < 3*4) \text{ OR } (2*3 > 2) = \text{True}$$

1 8 7 2 4 3 9 5 6

1.25 Pt

$$E2 = (y + \sqrt{z}/x) - ((2y/2 - 2)/2 - x)/y ; \text{ avec } x = 2, y = 6, z = 4$$

$$E2 = (6 + \sqrt{4}/2) - ((2*6/2 - 2)/2 - 2)/6$$

1

0.125 Pt

$$E2 = (6 + \sqrt{4}/2) - ((12/2 - 2)/2 - 2)/6$$

2

0.125 Pt

$$E2 = (6 + \sqrt{4}/2) - ((6 - 2)/2 - 2)/6$$

3

0.125 Pt

$$E2 = (6 + \sqrt{4}/2) - (4/2 - 2)/6$$

4

0.125 Pt

$$E2 = (6 + \frac{2}{2}) - (4/2 - 2)/6 \quad \text{0.125 Pt}$$

$$E2 = (6 + \frac{1}{6}) - (4/2 - 2)/6 \quad \text{0.125 Pt}$$

$$E2 = 7 - (4/2 - 2)/6 \quad \text{0.125 Pt}$$

$$E2 = 7 - (2 - 2)/6 \quad \text{0.125 Pt}$$

$$E2 = 7 - 0/6 \quad \text{0.125 Pt}$$

$$E2 = 7 - 0 \quad \text{0.125 Pt}$$

$$E2 = 7$$

2ème méthode :

$$E2 = (6 + \sqrt{4} / 2) - ((2 * 6 / 2 - 2) / 2 - 2) / 6 = 7 \quad \text{1.25 Pt}$$

6 4 5 10 1 2 3 7 8 9

5. Donner l'expression arithmétique correspondante à l'expression suivante écrite en Pascal : **(1 point)**

$$\exp(\text{sqr}(x)) / (2 * y - 1) + \text{abs}(x) - 1 / (\text{sqr}(x) + 3)$$

$$\frac{e^{\sqrt{x}}}{2y - 1} + |x| - \frac{1}{x^2 + 3} \quad \text{1 Pt}$$

6. Traduire l'expressions suivante en langage Pascal : **(1 point)**

$$E = |x| + \frac{e^{5x} + \sqrt{xy + x^2}}{3x + y^2}$$

$$E = \text{abs}(x) + \left(\exp(5 * x) + \text{sqr}(x * y + \text{sqr}(x)) \right) / (3 * x + \text{sqr}(y)) \quad \text{1 Pt}$$

Exercice 02 : [8 points]

Soit l'algorithme suivant :

```

Algorithme Exercice2 ;
Variables
  X, S, T : réels ;
  N, i : entiers ;
Début
  Lire (N, X) ;
  S ← 0 ; T ← - X*X ;
  Pour i ← 1 à N faire
    S ← S +  $\frac{T}{2i}$  ;
    T ← -T*X*X ;
  Fin-Pour ;
  Ecrire ('La somme est : ', S:0:1) ;
Fin.

```

1. Traduire l'algorithme donné en programme Pascal.
2. Dérouler l'algorithme donné pour X=1 et N= 3.
3. Dédire l'expression générale du résultat S en fonction de X et N.
4. Réécrire le programme Pascal en remplaçant la boucle **for** par la boucle **while** dans la partie de traitement.
5. Modifier l'algorithme pour calculer la somme S2 :

$$S2 = x^2 + \frac{x^4}{2} + \frac{x^6}{3} + \dots + \frac{x^{2N}}{N}$$

Corrigé de l'exercice 02 : [8 points]

1. Traduire l'algorithme donné en programme Pascal. (2 points)

Algorithme	Programme PASCAL
Algorithme Exercice2 ; Variables X, S, T : réels ; N, i : entiers ; Début Lire (N, X) ; $S \leftarrow 0$; $T \leftarrow -X * X$; Pour i $\leftarrow$ 1 à N faire $S \leftarrow S + \frac{T}{2i}$; $T \leftarrow -T * X * X$; Fin-Pour ; Écrire ('La somme est : ', S:0:1) ; Fin.	program Exercice2 ; 0.25pt Var X, S, T : real ; } 0.25pt N, i : integer ; Begin read (N, X) ; 0.25pt S := 0 ; T := - X*X ; {ou T := -sqr(X)} 0.25pt for i:=1 to N do 0.25pt begin {Begin et end} 0.25pt S := S + T/(2 * i) ; T := -T*X*X ; {ou T := -T*sqr(X)} } 0.25pt end ; write ('La somme est : ', S:0:1) ; 0.25pt End.

2. Dérouler l'algorithme donné pour X=1 et N= 3. (2 points)

Instructions	Variables					Affichage
	N	i	X	S	T	
Lire (N, X) ;	3		1			0.25 Pt
$S \leftarrow 0$; $T \leftarrow -X * X$; $T \leftarrow -1$	3		1	0	-1	0.25 Pt
Pour i $\leftarrow$ 1 $S \leftarrow S + \frac{T}{2i}$; $S \leftarrow 0 - \frac{1}{2}$; $S \leftarrow -0.5$ $T \leftarrow -T * X * X$; $T \leftarrow X * X * X * X$; $T \leftarrow 1$	3	1	1	-0.5	1	0.25 Pt
Pour i $\leftarrow$ 2 $S \leftarrow S + \frac{T}{2i}$; $S \leftarrow 0 - \frac{1}{2} + \frac{1}{4}$; $S \leftarrow -0.25$ $T \leftarrow -T * X * X$; $T \leftarrow -X * X * X * X * X * X$; $T \leftarrow -1$	3	2	1	-0.25	-1	0.5 Pt
Pour i $\leftarrow$ 3 $S \leftarrow S + \frac{T}{2i}$; $S \leftarrow 0 - \frac{1}{2} + \frac{1}{4} - \frac{1}{6}$; $S \leftarrow -0.41$ $T \leftarrow -T * X * X$; $T \leftarrow X * X * X * X * X * X * X * X$; $T \leftarrow 1$	3	3	1	-0.41	1	0.5 Pt 0.25 Pt
Écrire ('La somme est : ', S:0:1) ;	3	3	1	-0.41	1	La somme est : -0.4

3. D  duire l'expression g  n  rale du r  sultat S en fonction de X et N. **(1 point)**

Selon le d  roulement ci-dessus, nous avons :

Pour i = 1 , nous avons

$$S = -\frac{X^2}{2} = -\frac{1}{2} = -0.5$$

Pour i = 2, nous avons

$$S = -\frac{X^2}{2} + \frac{X^4}{4} = -\frac{1}{2} + \frac{1}{4} = -0.5 + 0.25 = -0.25$$

Pour i = 3, nous avons

$$S = -\frac{X^2}{2} + \frac{X^4}{4} - \frac{X^6}{6} = -\frac{1}{2} + \frac{1}{4} - \frac{1}{6} = -0.5 + 0.25 - 0.16 = -0.41$$

Pour i = N nous aurons :

$$S = -\frac{X^2}{2} + \frac{X^4}{4} - \frac{X^6}{6} + \dots \pm \frac{X^{2N}}{2N}$$

On peut g  n  raliser par la formule suivante :

$$S = \sum_{i=1}^N (-1)^i \frac{X^{2i}}{2i} \quad \text{ou} \quad S = -\frac{X^2}{2} + \frac{X^4}{4} - \frac{X^6}{6} + \dots \pm \frac{X^{2N}}{2N}$$

1 Pt

4. R   crire le programme Pascal en rempla  ant la boucle **for** par la boucle **while** dans la partie de traitement.

(1.5 point)

Programme PASCAL (avec la boucle While)

```

program Exercice2 ;
Var
  X, S, T : real ;
  N, i : integer ;
Begin
  read (N, X) ;
  S := 0 ; T := - X*X ; i:=1; {ou T := -sqr(X)}
  while (i<=N) do
    begin {begin et end}
      S := S + T/(2 * i) ;
      T := -T*X*X ; {ou T := - T*sqr(X)}
      i:=i+1 ;
    end ;
  write ('La somme est : ', S:0:1) ;
End.

```

0.25 Pt

0.5 Pt

0.25 Pt

0.5 Pt

5. Modifier l'algorithme pour calculer la somme S2 : (1.5 point)

Programme PASCAL (avec la boucle While)	
Algorithme Exercice2 ;	
Variables	
X, S, T : réels ;	
N, i : entiers ;	
Début	
Lire (N, X) ;	
S ← 0 ; T ← X*X ;	0.5 Pt
Pour i ← 1 à N faire	
S ← S + $\frac{T}{i}$;	0.5 Pt
T ← T*X*X ;	0.5 Pt
Fin-Pour ;	
Écrire ('La somme est : ', S:0:1) ;	
Fin.	

Exercice 03 : [4 points]

Une banque est ouverte de 8h00 à 12h00 et de 14h00 à 17h00, sauf le jeudi après-midi et le vendredi toute la journée. On suppose que l'heure h est un entier entre 0 et 23. Le jour j est également un entier de 1 à 7 (le code 1 pour samedi, 2 pour dimanche, etc).

Ecrire un programme Pascal qui demande le jour et l'heure, puis affiche si la banque est ouverte ou non.

Corrigé de l'exercice 04 :

Solution 1	
program Horaire_Banque;	0.25 Pt
Var	
h, j : integer ;	0.25 Pt
begin {begin et end}	0.25 Pt
write ('Donner un numéro de jour entre 1 et 7. Ex. 1 :samedi 2 : dimanche, ...');	
read (j) ;	0.25 Pt
write ('Donner l'heure, un entier entre 0 et 23');	
read (h) ;	0.25 Pt
if ((h>=8) and (h<=12) and (j<>7) or (h>=14) and (h<=17) and (j<6)) then	2 Pts
write('La banque est ouverte')	0.25 Pt
else	0.25 Pt
write ('La banque est fermée');	0.25 Pt
end.	

Solution 2

```
program Horaire_Banque;
Var
h, j : integer ;
b : Boolean ;
begin
    write ('Donner un numéro de jour entre 1 et 7. Ex. 1 :samedi 2 : dimanche, ...');
    read (j) ;
    write ('Donner l'heure, un entier entre 0 et 23') ;
    read (h) ;
    b := (h>=8) and (h<=12) and (j<>7) or (h>=14) and (h<=17) and (j<6);
    if (b=true) then
        write('La banque est ouverte')
    else
        write ('La banque est fermée') ;
    end.
end.
```

N.B : D'autres solutions sont possibles