

L'usage de la calculatrice est interdit

Examen de remplacement – Informatique 1

Exercice 01 : [8 pts]

1. Donner les définitions des termes suivants (deux lignes au maximum pour chaque terme) : Informatique, Algorithme.
2. Donner la structure de base d'un algorithme.
3. Evaluer les expressions suivantes en respectant l'ordre de priorité des opérateurs :
 $E1 = (a + d/c) - ((3d/2a + 8)/2 - c)/b$; avec $a = 8, b = 2, c = 2, d = 4$
 $E2 = (xy < z) \text{ OR NOT}(7 \text{ DIV } y^2 < x) \text{ AND}(y > 8x)$; avec $x = 6, y = 3, z = 1$
4. Traduire les expressions suivantes en langage Pascal :

$$E1 = x^4 + \frac{e^{4a} - \sqrt{a^2 + ab}}{2a + |2b|}, \quad E2 = 3a^2 + \sqrt{\frac{\sqrt{e^a} + b}{4a}}$$

5. Effectuer la conversion suivante :
 $(133)_8 = (\dots \dots)_{10}$

Exercice 02 : [7 pts]

Soit l'algorithme suivant :

```
Algorithme Exo2 ;  
Variables  
a, b, i, P : entier ;  
Début  
  Lire (a, b) ;  
  i ← 1 ;  
  Tantque (i ≤ b) faire  
    Si (a mod i = 0) et (b mod i = 0) Alors  
      P ← i ;  
    FinSi  
    i ← i+1 ;  
  Fin Tantque  
  Écrire ('P=',P) ;  
Fin.
```

1. Traduire l'algorithme donné en programme Pascal.
2. Dérouler l'algorithme donné pour a=16 et b= 4.
3. Déduire ce que fait l'algorithme ?
4. Réécrire l'algorithme en remplaçant la boucle **Tant-que** par la boucle **Pour** dans la partie de traitement.
5. Réécrire l'algorithme en remplaçant la boucle **Tant-que** par la boucle **Répéter** dans la partie de traitement.

Exercice 03 : [5 pts]

Écrire un programme Pascal qui demande à l'utilisateur de saisir une suite de valeurs entières. La saisie s'arrête lorsque l'utilisateur entre la valeur 0. Le programme doit calculer le nombre ainsi que la somme des éléments de la suite strictement croissante parmi les valeurs saisies. Le programme ne doit donc pas tenir compte de valeurs inférieures ou égales à la valeur précédemment acceptée (c-à-d la dernière plus grande valeur déjà saisie). Enfin, le programme affiche le nombre d'entiers pris en compte ainsi que leur somme.

Par exemple : pour la suite : -1 2 4 2 3 5 4 1 3 8 6 2 0

Le programme affichera : Nombre d'éléments : 5, somme : 18

Dans cet exemple, les valeurs 2 et 3 n'ont pas été prises en compte car elles étaient inférieures à 4. Pareil pour les trois valeurs 4, 1 et 3 car elles étaient inférieures à 5 et pareil pour les deux valeurs 6 et 2 car elles étaient inférieurs à 8.

Bonne chance

Corrigé de l'examen de remplacement – Informatique 1

Exercice 01 : [8 pts]

1. Donner les définitions des termes suivants : Informatique, Algorithme.

Les réponses possibles : (une seule définition requise)

Informatique : (0.5 point)

0.5

Définition 1 : l'informatique c'est le traitement automatique de l'information

Définition 2 : l'informatique est une branche qui s'occupe du domaine du traitement automatique de l'informatique.

Algorithme : (0.5 point)

Définition 1 : l'algorithme est un ensemble d'instructions séquentiellement et logiquement ordonnées, permettant de transformer des données en entrée en données de sortie, afin de résoudre un problème.

Définition 2 : Un algorithme est une série d'instructions détaillées qui décrivent comment effectuer une tâche ou résoudre un problème.

2. Donner la structure de base d'un algorithme. (0.75 point)

0.5

<i>Entête</i>	0.25	Algorithme NomAlgorithme
<i>Partie de déclaration</i>		Constante Identificateur=valeur Variable Identificateur : type
<i>Corps de l'algorithme</i>	0.25	Début Instruction 1 Instruction n
	0.25	Fin.

N.B : Si l'étudiant donne uniquement les trois parties (Entête, partie de déclaration et corps de l'algorithme) est aussi correct.

3. Evaluer les expressions suivantes en respectant l'ordre de priorité des opérateurs : (2.5 points)

$$E1 = (a + d/c) - ((3d/2a + 8)/2 - c)/b ; \text{ avec } a = 8, b = 2, c = 2, d = 4$$

6 5 10 12 3 4 7 8 9

- E1=(8+4/2)-((3*4/2*8+8)/2-2)/2 **0.125pt**
- E1=(8+4/2)-((12/2*8+8)/2-2)/2 **0.125pt**
- E1=(8+4/2)-((6*8+8)/2-2)/2 **0.125pt**
- E1=(8+4/2)-((48+8)/2-2)/2 **0.125pt**
- E1=(8+4/2)-(56/2-2)/2 **0.125pt**
- E1=(8+2)-(56/2-2)/2 **0.125pt**
- E1=10-(56/2-2)/2 **0.125pt**
- E1=10-(28-2)/2 **0.125pt**
- E1=10-26/2 **0.125pt**
- E1=10-13 **0.125pt**

1.25

N.B : Deux méthodes correctes :
 1 – Les différentes étapes
 2 – La numérotation et le résultat final

E1=-3

$E2 = (xy < z) \text{ OR } \text{NOT}(7 \text{ DIV } y * 2 < x) \text{ AND } (y > 8x)$; avec $x = 6, y = 3, z = 1$

1 2 10 8 3 4 5 9 7 6

- $E2 = (6*3 < 1) \text{ OR } \text{NOT}(7 \text{ DIV } 3*2 < 6) \text{ AND } (3 > 8*6)$ 0.125pt
- $E2 = (18 < 1) \text{ OR } \text{NOT}(7 \text{ DIV } 3*2 < 6) \text{ AND } (3 > 8*6)$ 0.125pt
- $E2 = \text{False OR NOT}(7 \text{ DIV } 3*2 < 6) \text{ AND } (3 > 8*6)$ 0.125pt
- $E2 = \text{False OR NOT}(2*2 < 6) \text{ AND } (3 > 8*6)$ 0.125pt
- $E2 = \text{False OR NOT}(4 < 6) \text{ AND } (3 > 8*6)$ 0.125pt
- $E2 = \text{False OR NOT True AND } (3 > 8*6)$ 0.125pt
- $E2 = \text{False OR NOT True AND } (3 > 48)$ 0.125pt
- $E2 = \text{False OR NOT True AND False}$ 0.125pt
- $E2 = \text{False OR False AND False}$ 0.125pt
- $E2 = \text{False OR False}$ 0.125pt

1.25

E2=False

4. Traduire les expressions suivantes en langage Pascal (2.25 point)

1.25

$$E1 := \underbrace{\exp(4 * \ln(x))}_{0.25} + \underbrace{(\exp(4 * a) - \text{sqrt}(\text{sqr}(a) + a * b))}_{0.5} / \underbrace{(2 * a + \text{abs}(2 * b))}_{0.25}$$

+0.25 Pour l'écriture linéaire

$$E2 := \underbrace{3 * \text{sqr}(a)}_{0.25} + \underbrace{\text{sqrt}((\text{sqrt}(\exp(a)) + b) / (4 * a))}_{0.5}$$

1

+0.25 Pour l'écriture linéaire

5. Effectuer les conversions suivantes : (1.5 point)

$$(133)_8 = (3 * 8^0 + 3 * 8^1 + 1 * 8^2)_{10}$$

$$(133)_8 = (91)_{10}$$

0.75

0.75

Exercice 02 : [7 pts]

1. Algorithme → Programme PASCAL (2 points)

Algorithme	Programme Pascal
<p>Algorithme Exo2 ;</p> <p>Variables a, b, i, P : entier ;</p> <p>Début Lire (a, b) ; i ← 1 ;</p> <p>Tantque (i ≤ b) faire Si (a mod i = 0) et (b mod i = 0) Alors P ← i ; FinSi ; i ← i+1 ;</p> <p>Fin Tantque Écrire ('P=',P) ;</p> <p>Fin.</p>	<p>Program Exo2 ; 0.125pt</p> <p>Var a, b, i, P : integer ; } 0.25pt</p> <p>Begin {Begin et End} 0.25pt read (a, b) ; 0.125pt i := 1 ; 0.125pt while (i <= b) do 0.25pt Begin {Begin et End} 0.25pt if (a mod i = 0) and (b mod i = 0) then 0.25pt P := i ; 0.125pt i := i+1 ; 0.125pt End ; write ('P=',P) ; 0.125pt End.</p>

2. Déroulement (2.5 points)

Instructions	Variables				Affichage	
	a	b	i	P		
Lire (a, b) ;	16	4	/	/	/	0.25
i ← 1 ;	16	4	1	/	/	0.25
<p>Tant-que (i ≤ b) faire (1 ≤ 4) vrai</p> <p>Si (16 mod 1 = 0) et (4 mod 1 = 0) Alors vrai vrai</p> <p> P ← i ; P ← 1 ;</p> <p>Fin-Si ;</p> <p> i ← i+1 ; i ← 1+1 =2;</p>	16	4		1	/	0.25
<p>Tant-que (i ≤ b) faire (2 ≤ 4) vrai</p> <p>Si (16 mod 2 = 0) et (4 mod 2 = 0) Alors vrai vrai</p> <p> P ← i ; P ← 2 ;</p> <p>Fin-Si ;</p> <p> i ← i+1 ; i ← 2+1 =3;</p>	16	4		2	/	0.25
<p>Tant-que (i ≤ b) faire (3 ≤ 4) vrai</p> <p>Si (16 mod 3 = 0) et (4 mod 3 = 0) Alors Faux Faux</p> <p> → je n'exécute pas le bloc Si → je passe à la prochaine itération</p> <p> i ← i+1 ; i ← 3+1 =4;</p>	16	4		2	/	0.25
<p>Tant-que (i ≤ b) faire (4 ≤ 4) vrai</p> <p>Si (16 mod 4 = 0) et (4 mod 4 = 0) Alors vrai vrai</p> <p> P ← i ; P ← 4 ;</p> <p>Fin-Si ;</p> <p> i ← i+1 ; i ← 4+1 =5;</p>	16	4		4	/	0.25
<p>Tant-que (i ≤ b) faire (5 ≤ 4) faux</p> <p> → je sors de la boucle Tant-que</p>	16	4	5	4	/	0.25
Écrire ('P=',P) ;	16	4	5	4	P=4	0.25

3. Dédire ce que fait l'algorithme ? (0.5 point)

0.5

L'algorithme calcule le PGCD de 2 entiers : PGCD (a,b)

4. Réécrire l'algorithme en remplaçant la boucle Tant que par la boucle Pour dans la partie de traitement. (1 point)

Algorithme avec la boucle Pour

Algorithme Exo2 ;

Variables

a, b, i, P : entier ;

Début

Lire (a, b) ;

{suppression de l'initialisation} 0.25pt

Pour i ← 1 à b faire 0.5pt

Si (a mod i = 0) et (b mod i = 0) Alors

P ← i ;

FinSi; {suppression de l'incrément} 0.25pt

FinPour ;

Écrire ('P=', P) ;

Fin.

5. Réécrire l'algorithme en remplaçant la boucle Tant que par la boucle Répéter dans la partie de traitement. (1 point)

Algorithme avec la boucle répéter

Algorithme Exo2 ;

Variables

a, b, i, P : entier ;

Début

Lire (a, b) ;

i ← 1 ;

Répéter 0.5pt

Si (a mod i = 0) et (b mod i = 0) Alors

P ← i ;

FinSi;

i ← i+1 ;

jusqu'à (i>b) ; 0.5pt

Écrire ('P=', P) ;

Fin.

Exercice 03 : [5 pts]

1 ^{ère} Solution possible	2 ^{ème} solution possible
<pre> Program exo3 ; 0.25pt Var Nbr, N, Somme, Max : integer ; } 0.5pt Begin {Begin et End} 0.25pt Write('Entrez un entier : '); Read(N) ; 0.25pt Max:=N ; 0.25pt Nbr:=1; 0.25pt Somme:=N; 0.25pt While (N<>0) do 0.5pt Begin 0.25pt write('Entrez un entier : '); Read(N) ; 0.25pt if (N>Max) then 0.5pt Begin {Begin et End} 0.25pt Max:=N; 0.25pt Somme:=Somme+Max ; 0.25pt nbr :=nbr+1 ; 0.25pt End; End; End; Write('Nombre delement : ', nbr, ' , Somme : ', Somme); 0.5pt End. </pre>	<pre> Program exo3 ; 0.25pt Var Nbr, N, somme, max : integer ; } 0.5pt Begin {Begin et End} 0.25pt Write('Entrez un entier : '); Read(N) ; 0.25pt Max:=N ; 0.25pt Nbr:=1; 0.25pt Somme:=N; 0.25pt Repeat 0.5pt write('Entrez un entier : '); Read(N) ; 0.25pt if (N>max) then 0.5pt Begin {Begin et End}0.25pt Max:=N; 0.25pt Somme:=Somme+Max ; 0.25pt nbr :=nbr+1 ; 0.25pt End; Until (N=0); 0.25pt Write('Nombre delement : ', nbr, ' , somme : ', somme); 0.5pt End. </pre>