

Examen

Durée : 01 heure 30 Minutes

N°Inscription :
 Nom :
 Prénom :
 Groupe :

Exercice 1 (7,5 Points) : Donner les valeurs des variables et l'affichage à l'écran après exécution des algorithmes suivants :

<p>1)</p> <p>Algorithme Algo1 ; Variable A, B, C, D :entier ; Début A ← 10 ; B ← 3 ; TantQue (A>B) Faire Si (A div B = B) Alors B ← A mod B ; A ← A mod B ; FinSi Si (B-A = B+ A) Alors C ← 10*B mod B +2 ; D ← C + (B - A) ; FinSi FinTantQue Ecrire(D+C- (B+A)) Fin.</p>	<p>2)</p> <p>Algorithme Algo2 ; Variable A, B, C, D :entier ; Début A ← 5 ; D ← 10 ; Pour B allant de 1 à 2 Faire Si (D - A = D div 2) Alors A ← A*2-D/2 ; C ← D mod 3 div 2 ; D ← D /2 div A ; Sinon A ← A-D ; C ← A div 2 ; Ecrire(A-C-D*2) ; FinSi FinPour Fin.</p>	<p>3)</p> <p>Algorithme Algo3 ; Variable A, B, C, D :entier ; Début A ← 10 ; B ← 1 ; C ← 8 ; Répéter Si (B>2) ou (A<9) Alors C ← (B + A) mod 5 ; Sinon D ← B + A div 3 ; FinSi A ← A-2 ; B ← B+1 ; Jusqu'à (A=C) et (B=D) ; Ecrire(A) ; Fin.</p>
---	--	---

0,5 * 15 (Points)	A	B	C	D	Écran
Algo1	0	1	2	3	4
Algo2	4	3	2	1	0
Algo3	4	4	4	4	4

Exercice 2 (3,5 Points) : Qu'affichent les programmes suivants ?

<p>1)</p> <pre>#include <iostream> using namespace std; int main() { int i, j; for(i=1; i<=5; i++) { for (j=1; j<=i; j++) { cout<<i; } cout<<"\n"; } return 0; }</pre>	<p>Écran</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <p>1 2 2 3 3 3 4 4 4 4 5 5 5 5 5</p> </div> <p><u>Triangle : 1 Point</u> <u>Valeurs : 1 Point</u></p>	<p>2)</p> <pre>#include <iostream> using namespace std; void Add (double &X , double Y) { X = X + 5; Y = X-7; } int main () { double X, Y; X=4 ; Y=7; Add(X, Y); cout<< "X = "<<X<<"\n"; cout<< "Y = "<<Y ; return 0 ; }</pre>	<p>Écran</p> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <p>X = 9 Y = 7</p> </div> <p><u>X=9 : 1 Point</u> <u>y=7 : 0,5 Point</u></p>
---	---	--	--

Exercice 3 (04 Points):

1) Traduire l'algorithme suivant en langage C++ en remplaçant la boucle **TantQue** par une boucle **Pour**

<p>Algorithme Exo3 ; Variable a, b, i, P : entier ; Début Lire (a, b) ; $i \leftarrow 2$; $P \leftarrow 1$; Tantque ($i \leq b$) faire Si ($a \bmod i = 0$) et ($b \bmod i = 0$) Alors $P \leftarrow i$; FinSi ; $i \leftarrow i+1$; FinTantque Ecrire(P) ; Fin</p>	<pre>#include <iostream> Using namespace std ; int main() { int a, b, i, p; <u>0,5 P</u> cin>>a>>b ; P=1 ; <u>0,5 P (0,25 *2)</u> for (i=2 ; i<=b ; i++) <u>1 P (0,25 : Structure + 0,25 *3 : Insts)</u> { if ((a % i ==0) && (b % i ==0)) <u>0,5 P (le bloc if)</u> p=i ; } cout<<p ; <u>0,5 P (cout + main() + les accolades)</u> return 0 ; }</pre>
---	---

2) Que fait ce programme ? Calcule le PGCD de 2 entiers : PGCD (a,b) **(01 Point)**

Exercice 4 (05 Points) : Pour calculer la **Partie Entière** de la **Racine Carrée** (PERC) d'un nombre entier, on utilise la méthode suivante :

$$\begin{aligned}
 1 &= 1^2 \\
 1+3 &= 4 = 2^2 \\
 1+3+5 &= 9 = 3^2 \\
 1+3+5+7 &= 16 = 4^2 \text{ etc...}
 \end{aligned}$$

Exemple : PERC (9) = 3 ; PERC (16) = 4 ; **PERC (15) = 3 (Partie entière de $\sqrt{15} = 3,87$ et 3)**

Écrire l'algorithme qui permet de lire un entier strictement positif et de calculer la partie entière de sa racine carrée. En utilisant une procédure pour lire le nombre entier strictement positif et une fonction pour le calcul de la racine carrée (avec la méthode ci-dessus)

<p>Algorithme Racine_Carre ; Variable N : Entier ; (0,25 Point) Procédure Lire_Entier (0,25 Point) Début Répéter (Boucle : 0,5 Point) Ecrire(" Donner un entier positif") ; Lire(N) ; (0,25 Point) Jusqu'à (N>0) ; Fin ; Fonction PERC (X : Entier) : Entier (0,25 Point) Variable i, S, R : entier ; (0,25 Point) Début $i \leftarrow 1$; $S \leftarrow 0$; $R \leftarrow 0$; (0,5 Point) TantQue (S < X) Faire (0,5 Point) $i \leftarrow i+2$; $S \leftarrow S+i$; $R \leftarrow R+1$; (0,75 Point) FinTanque</p>	<pre>PERC ← R ; <u>(0,5 Point)</u> Fin ; Début Lire_Entier ; <u>(0,5 Point)</u> Ecrire (PERC (N)) ; <u>(0,5 Point)</u> Fin.</pre>
--	--