Cette évaluation va permettre de tester votre compréhension des algorithmes et va vous laisser écrire vous-même vos algorithmes.

|  |  |
| --- | --- |
| Compréhension : | 5 pts |

## Question 1 (1 point)

La déclaration des variables a été effacée dans mon algorithme. Pouvez-vous la réécrire en déduisant les types des variables a et b d’après leurs utilisations ?

Variables :

a : chaîne de caractère

b : entier

Début

Saisir a

Saisir b

Si a = « beau » et b > 15 alors

Afficher « Le temps est beau »

Sinon si a <> « beau » et b < 15 alors

Afficher « Pas génial »

Sinon

Afficher « ni moche ni beau »

Fsi

Fin

## Question 2 (1 point)

Dans l’algorithme ci-dessous j’ai effacé par inadvertance la fin du message à afficher à l’utilisateur. Pouvez le retrouver en fonction de ce que fait l’algorithme ?

Variables :

Entier[] : t

Début

t = [2, 5, 7]

si t[0] =< t[1] et t[1] =< t[2] alors

Afficher « Les 3 éléments de la liste sont croissants »

fsi

Fin

## Question 3 (1 point)

Je définis la fonction suivante :

Fonction inconnu(entier : a, entier : b, entier : c) : entier

Début

Si a < b et a < c alors

Retourne b \* c

Sinon si b < a et b < c alors

Retourne a + c

Sinon

Retourne a – b

Fsi

Fin

Que retourne cette fonction si je l’appelle avec les paramètres suivants :

Resultat 🡨 inconnu(7, 5, 1)

L’algorithme va retourner « 2 »

Il utilise le deuxième « sinon » car « a » est plus grand que « b » et plus grand que « c » et « b » est plus grand que « c ». Il ne reste donc que la troisième option, retourner « a – b ».

## Question 4 (1 point)

Si je définis un tableau d’entiers de la manière suivante :

Entier[] : t = [17, 25, 6, 7, 85, 9, 12]

Que vaut t[3] ?

t[3] vaut « 7 » car il est le 4ème terme de la liste après « t[0] = 17 », « t[1]=25 » et « t[2]=6 »

## Question 5 (1 point)

Si je définis un tableau d’entiers de la manière suivante :

Entier[] : t = [[17, 25, 6],[ 7, 85, 9], [12, 54, 27]]

Que vaut t[1][0] ?

t[1][0] vaut « 7 »

|  |  |
| --- | --- |
| Lecture et affichage : | 5 pts |

## Question 1 (2.5 points)

Ecrire un algorithme qui demande à un utilisateur de saisir deux nombres entiers et affiche le produit des nombres.

|  |
| --- |
| Variables  a : entier  b : entier  Début  saisir a  saisir b  afficher a \* b  Fin |

## Question 2 (2.5 points)

Ecrire un algorithme qui demande à un utilisateur de saisir trois nombres entiers et affiche le plus petit des nombres.

|  |
| --- |
| Variables  a : entier  b : entier  c : entier  Début  saisir a  saisir b  saisir c  si a < b et a < c  alors écrire a  sinon si b < a et b < c  alors écrire b  sinon  écrire c  Fsi  Fin |

|  |  |
| --- | --- |
| Boucles : | 5 pts |

## Question 3 (2.5 points)

Ecrire un algorithme qui affiche tous les nombres multiples de 7 entre 0 et 100.

|  |
| --- |
| Variables  T : entier[]  Multiple : entier[]  Début  T 🡨 [0,1,2…100]  Pour i de T[0] à T[101]  Si i % de 7 = 0  Multiple 🡨i  Fsi  Afficher Multiple  FinPour  Fin |

## Question 4 (2.5 points)

Ecrire un algorithme qui demande à un utilisateur de saisir un nombre pair positif et qui redemande la saisie tant que le nombre n’est pas pair et positif.

|  |
| --- |
| Variables  a : entier  Début  Saisir a  Si a %2==0 et si a >=0  écrire « ok »  Sinon  écrire « Veuillez écrire un nombre pair ET positif »  Fsi  Fin |

|  |  |
| --- | --- |
| Fonctions : | 5 pts |

## Question 5 (2.5 points)

Ecrire une fonction qui prend en paramètre un tableau d’entier et qui retourne le nombre d’éléments dans le tableau qui sont supérieurs à 20.

|  |
| --- |
| Fonction Supérieur (l:entier[]):entier  variables locales :  a : entier []  Début  pour i de l  si i > 20  a <- i  Fsi  FinPour  retourner a  Fin |

## Question 6 (2.5 points)

Ecrire une fonction qui prend en paramètre un nombre entier et qui détermine si ce nombre correspond à une année bissextile. Une année bissextile est divisible par 4 mais pas par 100 ou alors par 400.

|  |
| --- |
| Fonction Bissextile (a:entier):Booléen  Variables locales :  b : Booléen  Début  Pour i de a  Si a %400==0 ou si a%4==0  B <- True  Sinon si a%100==0  B <- False  Fsi  Retourner b  FinPour  Fin |