

TD ALGORITHMIQUE 002

DEV-DATA P6

Partie 3 : Structures itératives

Exercice 1 : Un nombre est parfait s'il est égal à la somme de ses diviseurs stricts (différents de lui-même). Ainsi par exemple, l'entier 6 est parfait car $6 = 1 + 2 + 3$. Écrire un algorithme permettant de déterminer si un entier naturel est un nombre parfait.

Exercice 2 : Écrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on tape 4, l'algorithme doit calculer: $1 + 2 + 3 + 4 = 10$ Réécrire l'algorithme qui calcule cette fois la moyenne !

Exercice 3: Faire un programme qui calcule et affiche la division de a par b par soustractions successives

Exercice 4: Faire un programme qui calcule le PGCD de deux nombres saisis au clavier en utilisant l'astuce suivante: soustrait le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

Exercice 5: Faire un programme qui calcule et affiche le PPCM de deux entiers saisis au clavier.

Exercice 6 : Écrire l'algorithme qui affiche la somme des prix d'une suite d'articles en CFA (entiers) saisies par l'utilisateur et se terminant par zéro.

Exercice 7 : Écrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui affiche à la fin le plus grand de ces 10 nombres Et affiche aussi son rang dans la liste saisie

Exercice 8 : Écrire un algorithme mettant en œuvre le jeu suivant entre deux joueurs : Le premier utilisateur saisit un entier que le second doit deviner. Pour cela, il a le droit à autant de tentatives qu'il le souhaite. A chaque échec, le programme lui indique si l'entier est plus grand ou plus petit que sa proposition. Un score est affiché lorsque l'entier est trouvé.

Exercice 9 : On se propose de saisir N entiers différents entre 1 et 100 (N étant un entier naturel compris entre 10 et 50) puis afficher la plus longue séquence croissante tout en précisant la position du premier nombre de cette séquence.

Exercice 10: Écrire un programme qui permet de tester si deux nombres sont AMIS ou pas. Deux nombres M et N sont amis si la somme des diviseurs de N excepté 1 et lui-même est égale à M et la somme des diviseurs de M excepté 1 et lui-même est égale à N.

Exemple: les nombres 48 et 75 sont deux nombres amis puisque :

Les diviseurs de 48 sont: 2, 3, 4, 6, 8, 12, 16, 24 $\Rightarrow 2 + 3 + 4 + 6 + 8 + 12 + 16 + 24 = 75$

Les diviseurs de 75 sont : 3, 5, 15, 25 $\Rightarrow 3 + 5 + 15 + 25 = 48$

Exercice 11: Écrire un programme qui permet de saisir un nombre entier positif, le programme détermine et affiche si le nombre est premier ou pas.

Exercice 12: Écrire un programme qui permet de saisir un nombre entier positif, le programme détermine et affiche si le nombre est parfait ou pas.

Exercice 13: Écrire un programme qui permet de saisir une suite de nombres premiers, le programme s'arrête lorsque l'utilisateur tape la valeur zéro (0). Le programme affiche le nombre d'entiers saisi, le nombre de nombre premiers et affiche tous les nombre premiers.

Exercice 14: Écrire un programme qui permet de saisir un entier positif N, le programme affiche tous les nombres premiers compris entre 1 et N.

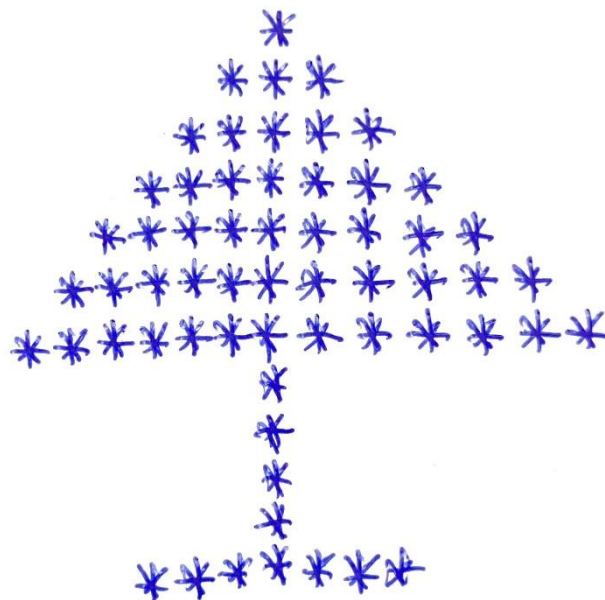
Exercice 15: Écrire un programme qui permet de saisir un entier positif N, le programme affiche les n premiers nombre premiers

Exercice 16: Écrire un programme qui permet de dessiner un sapin. le programme demande les informations suivantes avant d'afficher le sapin :

- le symbole à utiliser pour dessiner le sapin : \$, *, + ou 0.
- la hauteur des feuilles.
- la hauteur du tronc.
- la largeur des racines.

Exemple:

symbole : * hauteur feuilles : 7 hauteur tronc: 4 largeur racine : 7



Partie 4 : Fonctions et Procédures

Reprendre les exercices ci-dessous avec les fonctions

Exercice 1 : Faire un sous-programme qui reçoit une année puis indique si l'année est bissextile ou pas.

Exercice 2 : Faire un sous-programme qui reçoit une année et un mois puis détermine et affiche le nombre de jours de ce mois dans cette année .

Exercice 3 : Faire un sous-programme qui reçoit une date (jour, mois et année) puis indique si la date est valide ou pas.

Exercice 4 : Faire un sous-programme qui reçoit qui saisit une date (jour, mois et année) puis détermine et affiche la date suivante.

Exercice 5 : Faire un sous-programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date précédente.

Exercice 6 : Faire un sous-programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date qu'il fera dans 5 jours.

Exercice 7 : Faire un sous-programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date qu'il faisait il y a N jours est saisi au clavier et est positif.

Exercice 8 : Faire un sous-programme qui reçoit une un nombre positif puis détermine la somme des diviseurs de ce nombre.

Exercice 9 : Écrire un sous-programme qui permet de tester si deux nombres sont AMIS ou pas. Deux nombres M et N sont amis si la somme des diviseurs de N excepté 1 et lui-même est égale à M et la somme des diviseurs de M excepté 1 et lui-même est égale à N.

Exemple : les nombres 48 et 75 sont deux nombres amis puisque :

Les diviseurs de 48 sont : 2, 3, 4, 6, 8, 12, 16, 24 => $2 + 3 + 4 + 6 + 8 + 12 + 16 + 24 = 75$

Les diviseurs de 75 sont : 3, 5, 15, 25 => $3 + 5 + 15 + 25 = 48$

Exercice 10 : Faire un sous-programme qui reçoit une un nombre positif puis détermine si nombre est parfait ou pas.

Exercice 11 : Faire un sous-programme qui reçoit une un nombre positif puis détermine si nombre est premier ou pas.

Exercice 12 : Écrire un programme qui contient les sous-programmes suivants :

- Un sous-programme qui permet de saisir un nombre positif
- Un sous-programme qui saisit une série de nombre positifs, la saisie s'arrête lorsque l'utilisateur tape la valeur zéro (0). Ce sous-programme détermine et affiche :
 - Le nombre d'entiers saisi
 - Le nombre de nombre premiers
 - Tous les nombres premiers.
 - Produit des nombres parfaits

Exercice 13 : Écrire un programme qui permet de saisir un entier positif N , le programme affiche tous les nombres premiers compris entre 1 et N . Ce programme doit être subdivisé en sous-programme.

Exercice 14 : Écrire un programme qui permet de saisir un entier positif N , le programme affiche les n premiers nombre premiers. Ce programme doit être subdivisé en sous-programme.

Exercice 15 :

On se propose de saisir N entiers différents entre 1 et 100 (N étant un entier naturel compris entre 10 et 50) puis afficher la plus longue séquence croissante tout en précisant la position du premier nombre de cette séquence. Ce programme doit être subdivisé en sous-programme.