**TD Programmation ABOU NIANG**

**Exercice 1 :**

Ecrire un programme qui saisit deux entiers a et b, calcule et affiche le quotient entier, le reste de la division et le ratio (quotient réel).

SOLUTION

**Exercice 2**

Ecrire un programme qui demande à l’utilisateur de donner le rayon d’un cercle et lui retourne sa surface et son périmètre.

PI = 4 \* arc tangente de 1. La fonction arc tangente est atan ex : atan(2).

SOLUTION

ProgrammCercle

Exo 2 :

Programme Cercle

   const pi = 4 \* atan(1)

 var r : entier

DEBUT

Ecrire (“ veuillez saisir le rayon r du cercle”)

Lire r

Ecrire (“la surface du cercle est :”, pi \* r \* r , “le périmètre du cercle est  :”, 2 \* pi  \*  r )

  FIN

**Exercice 3 :**

Version 1 :

Faire un programme qui saisit 3 résistances : R1, R2 et R3.

Calculer et afficher la résistance en série : R1 + R2 +R3

Calculer et afficher la résistance en parallèle : (R1 \* R2 \* R3) / (R1\*R2 + R2\*R3 + R1\*R3)

**Exo 3 Solution Version 1**

Programme CalculDeResiStance

Var R1 , R2, R3 : reel

DEBUT

Ecrire (“Veuillez saisir les resistances R1 , R2 , et R3”)

lire R1 , R2 , R3

Ecrire (“la resistance en serie est : ” , R1 + R2 + R3 )

Ecrire(“la resistance en derivation est ; ”, (R1 \* R2 \* R3) /(R1\*R2 + R2\*R3 + R1\*R3) )

FIN

**Exercice 3 Version 2** :

Demander à l’utilisateur d’indiquer son choix.

S’il entre la valeur 1, calculer et afficher la fréquence en série.

S’il entre la valeur 2, calculer et afficher la fréquence en parallèle.

**Exo 3 Solution Version 2**

Programme CalculDeResistance

var R1 , R2 , R3 : reel

var x : entier

DEBUT

Ecrire(“veuillez entrer les resitances R1, R2 et R3 ”)

lire R1 , R2 , R3

Ecrir(“veuillez entrer un nombre x”)

Lire x

Si x = 1

Alors Ecrire(“la resistance en serie est :”, R1 + R2 + R3)

sinon si x = 2

                     Alors Ecrire (“la resistance en parallele est : ”(R1 \* R2 \* R3)/(R1\*R2 + R2 \* R3 + R1\*R3))

              Finsi

     Finsi

Fin

**Exercice 4**

Ecrire un programme qui saisit un réel x et un entier n et affiche x à la puissance n.

Version 1 : utiliser la fonction pow du fichier d’en-tête <math.h> ex : pow(x,n)

Version 2 : en utilisant un boucle

**Solution**

**ProgrammePuissance**

Var x , puissance : reel

var i , n  : entier

Debut

Ecrire (“veuillez saisir un reel x et un entier n ”)

lire  x , n

 puissance ← 1

Pour i ← 1 à n Faire

 Puissance ←  puissance \* x

Finpour

Ecrire (“x à la puissance n est :” , puissance)

Fin

Exercice 5 :

Ecrire un programme qui saisit 5 variables de type entier au clavier et qui affiche leur somme. Utiliser une boucle (for ou while ou do..while).

Solution

ProgrammeSomme

var i , total , val : entier

const arret = 5

Debut

Total ← 0

I 🡨 1

Tant que i <= arret faire

Ecrire (“Veuillez saisir un une valeur ”)

Lire val

total← total + val

i 🡨 i +1

FinTant que

Ecrire (“la somme des 5 nombres saisi est ” total)

Fin

**Exercice 6 :**

Faire un programme qui saisit les coordonnées de 2 points A (x1, y1) et b(x2, y2) et qui affiche la distance entre les 2 points.

Formule : distante = racine carrée de ((x1 – x2)2 + (y1 – y2)2)

Racine carrée : sqrt. Ex : sqrt(7) ; <math.h>

Solution

Programme coordonner

var x1 , x2 , y1 , y2 : reel

 var abscisse , ordonner =  reel

Debut

 Ecrire (“veuillez donner les coordonne du point A”)

lire ( x1 , y1)

Ecrire (“veuillez ecrie les coordonner du poinr B”)

  lire ( x2 , y2)

abscisse  ←  (x1-x2)\*(x1-x2)

ordonner  ← (y1-y2)\*(y1-y2)

 Ecrire (“la distance des deux point A et B est” , sqrt (abscisse + ordonner) )

 Fin

Exercice 7 :

Décomposition d’un montant en euros Écrire un algorithme permettant de décomposer un montant entré au clavier en billets de 20, 10, 5 euros et pièces de 2, 1 euros, de façon à minimiser le nombre de billets et de pièces.

**Solution**

**Programme Decomposition**

var C20 , C10 , C5 , p2 , p1 , N : entier

Ecrire (« veuillez saisir un montant en euro »)

 lire N

C20← N div 20

C10← (N mod 20) div 10

C5← ((N mod 20) mod 10) div 5

p2← (((N mod 20) mod 10)  mod5) div 2

p1← ((((N mod 20) mod 10)  mod5) mod2) div 1

Ecrire (“Le montant de ” , N , “euros saisit est équicalent à ” , c20 , “euro” , c10 ,”euro” , c5 , “euro” , p2 , “piéce et de”, p1 , “piéce” )

Fin .

**Exercice 8** :

Ecrire un algorithme permettant de résoudre une équation du second degré.

Ax2 + bx + c = 0

**Solution**

var a , b , c : reel

var delta  :reel

Debut

Ecrire (“Veuillez entrer les coéfficient a , b , et c de l’equation du degrés degrés ax2 + bx + c”)

 lire a , b , c

 delta ← (b\*b) - (4\*a\*c)

 Si  delta < 0

Sinon Si delta  > 0

  Alors Ecrire(“l’equation admet deux solutions ” , (-b+sqrt(delta))/2a , “et” , (-b-sqrt(delta))/2a)

   Sinon Si delta = 0

 Alors Ecrire(“l’equation admet une solution double” , -b  /2a)

                         FinSi

  FinSi

FinSi

Fin .

**Exercice 9** :

Ecrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée

a. On considère que le départ et l'arrivé ont lieu le même jour

**Solution**

**Programme DuréeDuVoyage**

**var ha , mna , hd , mnd : entier**

**Debut**

**Ecrire (“Veuillez l’heure et la minute de votre depart”)**

**Lire hd , mnd**

**Ecrire (“Veuillez entre l’heure et la minute de votre arrivé”)**

**lire ha , mna**

**Si (mna < mnd)**

**Ecrire (“votre heure d’arrové est :”, ha-(1 + hd) , “h” , (mna + 60- mna),”minute”)**

**Sinon Ecrire(“votre heure d’arrivé est :”, ha - hd , “h” , mna - mnd,”minute”)**

**Finsi**

**Fin**

**Exercice 9 :**

**Ecrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée**

b. On suppose que la durée de vol est inférieure à 24 heures mais peut avoir lieu le lendemain.

**Solution**

**Programme DuréeDuVoyage**

**Var ha , mna , hd , mnd :entier**

**Debut**

**Ecrire(“Veuillez l’heure et la minute de votre depart”)**

**lire hd , mnd**

**Ecrire (“Veuillez entre l’heure et la minute de votre arrivé”)**

**lire ha , mna**

**Si (mna < mnd)**

Si (hd>ha )

  AlorsEcrire(“Votre voyage a duré :” 23-hd+ha , “heure et ”, mna  + 60-mnd,”minute”)

   Sinon Si (hd<ha et  mnd<mna)

   AlorsEcrire(“votre voyage à durée ”, ha-hd ,”heures et ”,mna-mnd,”minutes”)

   Sinon Si(hd<ha et mnd>mna)

 AlorsEcrire(“votre voyage à durée” , ha - (1 + hd) ,”heure et” , mna + 60 - mnd )

 Finsi

             Finsi

  Finsi

Fin

**Exercice 10 :**

Ecrire un algorithme qui lit trois valeurs entières ( A, B et C) et qui permet de les trier par échanges successifs Et enfin les afficher dans l'ordre 4

**Solution**

**Programme par tri succésive**

**Var tableau t(3) : entier**

**var i , j , min :  entier**

**Debut**

**Ecrire (<< veuillez entrer trois entiers distinct A , B , C >>)**

**Lire t(1) , t(2) , t(3)**

**Pour i allant 2 à 3 faire**

**min ← t[i]**

**j ← i-1**

**Tant que (j>=1 et t[j]>min) faire**

**t[j+1]<-- t[j]**

**j ← j-1**

**FinTantque**

**t[j+1]<-- min**

**FinPour**

**Fin**

**Exercice 11 :**

Ecrire un algorithme calculatrice permettant la saisie du premier entier (a) de l'opération ( + ou – ou \* ou / : sont des caractères) et du deuxième entier (b) et qui affiche le résultat.

**Solution**

**Programme Calculatrice**

**var a , b : entier**

**var ope : caractere**

**Debut**

**Ecrire (“veuillez saisire le premier entier ”)**

**lire a**

**Ecrire (“veuiller saisir l’operateur ”)**

**lire ope**

**Ecrire (“veuillez saisir le deuxiéme entier”)**

**lire b**

**Selon Que**

**ope = + :Ecrire (“la somme de ”, a , “et”, b, ”est :” , a+b )**

**ope = \* :Ecrire (“la multilication de “, a , “et” ,b,”est :”, a\*b)**

**ope = - :Ecrire (“la soustraction de ”, a , ”et”, b, “est” a-b)**

**ope =/ et si (b!=0) :Ecrire(“la division de ”, a , “et”, b ,”est :”, a/b)**

**sinon :Ecrire (“l’opération est invalide”)**

**Fsi**

**Sinon Ecrire (<< l’operation est invalide >>)**

**Fin Selon Que**

**Fin**

**Exercice 12 :**

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

**Solution**

ProgrammeNombreParfait

var  N , i , som : entier

Debut

repeter

Ecrire(“Veuillez entrer un entier N”)

 lire N

jusqu’a (N > 0)

som ← 0

Pour i ← 1 à (N div 2) faire

  r ← N mod i

si (r = 0) Alors

som ← som + r

FinSi

FinPour

Si (som = N)

 Alors Ecrire (“le nombre” , N , “est un nombre parfait ”)

Sinon Ecrire (“le nombre” , N ,  “n’est pas parfait”)

FinSi

Fin

**Exercice 13 :** Faire un programme qui saisit une date (jour, mois et année) at qui indique si la date est valide

Programme ProgrammeCalendrier

var a , m, j , : entier

Debut

Répéter

Afficher (“Saisir jour, mois et année ”)

Lire (j, m,a )

jusqu’à (j <= 31 ET m <= 12 )

Si (j= 30 ET m=2 ou j = 31 et m=2)

alors Ecrire (“la date est invalide “)

Sinon Afficher (“La date est valide”)

FinSi

Fin