Resolution Exercice 1

**Programme calcul**

var a,b,q,r,ratio: entier

**Debut**

Afficher(«Saisir a et b:  »)

Saisir(a,b)

si(b!=0)

Alors

q ← a **div** b

r ← a **mod** b

ratio → a/b

sinon

Afficher(«Division impossible»)

fsi

Afficher(«Le quotient entier est :»,q)

Afficher(«Le reste est :»,r)

Afficher(«Le quotient reel est :»,ratio)

**Fin**

Resolution Exercice 2

**Programme calculRayon**

const PI = 4\*atan(1)

var r,s,p: entier

**Debut**

Afficher(«Saisir un rayon:  »)

Saisir(r)

s ← PI\*r\*r

p ←2\*PI\*r

Afficher(«La surface est :»,s)

Afficher(«Le périmètre est :»,p)

**Fin**

Resolution Exercice 3 version1

**Programme calculRegistance**

var r1,r2,r3 : reel

rp,rs : reel

**Debut**

Afficher(«Saisir les 3 registance :  »)

Saisir(r1,r2,r3)

rs ← r1+r2+r3

rp ← (r1\*r2\*r3)/(r1\*r2 + r2\*r3 + r1\*r3)

Afficher(«Registance serie :»,rs)

Afficher(«Registance paralle :»,rp)

**Fin**

Resolution Exercice 3 version2

**Programme calculRegistance**

var r1,r2,r3 : reel

rp,rs : reel

choix : eniter

**Debut**

Afficher(«Saisir les 3 registance :  »)

Saisir(r1,r2,r3)

Afficher(«Taper 1 pour calcule frequence en serie ou 2 pour paralle :  »)

Afficher(«Saisir votre choix  »)

Saisir(choix)

**suivant**(choix) faire

case1 :

rs ← r1+r2+r3

Afficher(«Registance serie :»,rs)

case2 :

rp ← (r1\*r2\*r3)/(r1\*r2 + r2\*r3 + r1\*r3)

Afficher(«Registance paralle :»,rp)

default :

Afficher(«Aurevoir !!!!!!!!»)

**Fin**

Resolution Exercice 4 v1

**Programme calculPuissance**

var x,n,p : entier

**Debut**

Afficher(«Saisir x et n :  »)

Saisir(x,n)

p ← pow(x,n)

Afficher(x,«a la puissance  ,n ,« est:»,p)

**Fin**

Resolution Exercice 4 v2 avec boocle for

**Programme calculPuissance**

var x,n,p,i : entier

**Debut**

Afficher(«Saisir x et n :  »)

Saisir(x,n)

p ← 1

**pour** i allant de 1 a n faire

p ← x\*p

**fpour**

Afficher(x,«a la puissance  ,n ,« est:»,p)

**Fin**

Resolution Exercice 5

**Programme calculSomme**

var x,i,som: entier

**Debut**

som← 0

**pour** i allant de 1 a 5 faire

Afficher(«Saisir un nombre :  »)

Saisir(x)

som← som+x

**fpour**

Afficher(«La somme des 5 variable saisie est :  »,som)

**Fin**

Resolution Exercice 6

**Programme calculDistance**

var x1,x2,y1,y2: entier

distance : entier

**Debut**

Afficher(«Saisir les coordoné de A(x1,y1)»)

Saisir(x1,y1)

Afficher(«Saisir les coordoné de B(x2,y2)»)

Saisir(x2,y2)

distance← sqrt((x1-x2)\*2+(y1-y2)\*2)

Afficher(«La distance est :  »,distance)

**Fin**

Resolution Exercice 7

**Programme DecomposMontant**

var billet20, billet10, billet5 : entier

piece2, piece1: entier

rest,montant : entier

**Debut**

Afficher(«Saisir le montant a decomposer»)

Saisir(montant)

billet20← montant div 20

rest← montant mod 20

billet10← rest div 10

rest← rest mod 10

billet5← rest div 5

rest← rest mod 5

piece2← rest div 2

rest← rest mod 2

piece1 → reste

Afficher(«Les billet et piece decomposé sont :  »,biller20,billet10,billet5,piece2,piece1)

**Fin**

Resolution Exercice 8

**Programme Equation**

var a,b,c: entier

deltat,x,x1,x2: entier

**Debut**

Afficher(«Saisir les valeur de a b c de l ‘équation ax2+bx-c=0»)

Saisir(a,b,c)

deltat = b\*b-4\*a\*c

**si**(deltat < 0)

**alors**

Afficher(«Pas de solution»)

**fsi**

**si**(deltat == 0)

**alors**

x→ -b/2\*a

Afficher(«La solution unique est :»,x)

**fsi**

**si**(deltat > 0)

**alors**

x1→ - b -sqrt(deltat)/2\*a

x2→ - b +sqrt(deltat)/2\*a

Afficher(«X1= »,x1, «X2= :»,x2)

**fsi**

**Fin**

Resolution Exercice 9

**Programme calculDurreVol**

***var***  heure\_depart, heure\_arrive, minute\_depart, minute\_arrive : entier

dure\_heur, dure\_minute : entier

**Debut**

Afficher(«Saisir les horaire de depart : »)

Saisir(heure\_depart, minute\_depart)

Afficher(«Saisir les horaire de d’arrive : »)

aisir(heure\_arrive, minute\_arrive)

**SI**(heure\_arrive > heure\_depart )

//L’hypothése a

**Alors**

**SI**(minute\_arrive>minute\_depart)

**Alors**

dure\_heur→ heure\_arrive – heure\_depart

dure\_minute → minute\_arrive – minute\_depart

Afficher(«La durée de vol est : », dure\_heur, dure\_minute)

**Sinon**

dure\_heur→ heure\_arrive – heure\_depart - 1

dure\_minute → minute\_arrive + 60 – minute\_depart

Afficher(«La durée de vol est : », dure\_heur, dure\_minute)

**FSI**

**Sinon**

**//L’hypothése b**

**SI**(minute\_arrive>minute\_depart)

dure\_heur→ heure\_arrive – heure\_depart+24

dure\_minute → minute\_arrive – minute\_depart

Afficher(«La durée de vol est : », dure\_heur, dure\_minute)

**Sinon**

dure\_heur→ heure\_arrive – heure\_depart + 24 – 1

dure\_minute → minute\_arrive + 60 – minute\_depart

Afficher(«La durée de vol est : », dure\_heur, dure\_minute)

**FSI**

**Fsi**

**Fin**

Resolution Exercice 10

**Programme TriNombre**

**procedure permut(x:entier,y:entier)**

**var tmp : entier**

**Debut**

**tmp→ x**

**x→ y**

**y→ tmp**

**Fin**

var a,b,c : entier

**Debut**

Afficher(«Saisir a,b,c:  »)

Saisir(a,b,c)

***SI***(a>b)

Alors

//apelle de notre function permut

permut(a,b) //a est plus petit que b

***SI***(b>c)

Alors

permut(b,c) //c est plus petit que b

// je teste a nouveau la valeur de a et b

***SI***(a>b)

Alors

permut(a,b) //a est plus petit que b

**FSI**

**FSI**

**Sinon**

***SI***(b>c)

Alors

permut(b,c) //b est plus petit que b

// je teste a nouveau la valeur de a et b

***SI***(a>b)

Alors

permut(a,b) //c est plus petit que b

**FSI**

**FSI**

**FSI**

Afficher(«Les valeur apres tri :»,a,b,c)

**Fin**

**Resolution Exercice 11**

**Programme Calculatrice**

var a,b :entier

op: caracter

**Debut**

Afficher(«Saisir l’entier a»)

Saisir(a)

Afficher(«Faire une operation»)

Saisir(op)

Afficher(«Saisir l’entier b»)

Saisir(b)

**SI**(op ==’ +’)

**Alors**

Afficher(«La sommme est :  » ,a+b)

**FSI**

**SI**(op ==’ -’)

**Alors**

Afficher(«La differance est :  » ,a-b)

**FSI**

**SI**(op ==’ \*’)

**Alors**

Afficher(«Le produit est :  » ,a\*b)

**FSI**

**SI**(op ==’ /’)

**Alors**

**SI**(b !=’0)

**Alors**

Afficher(«Ops !!! division impossible  » )

**Sinon**

Afficher(«Le quotient est :  » ,a/b)

**FSI**

**FSI**

**Fin**

Resolution Exercice 12

**Programme TestNombreParfait**

var nbr,i,som: entier

**Debut**

som← 0

Afficher(«Saisir un nombre :  »)

Saisir(nbr)

**pour** i allant de 1 a nbr-1 faire

**SI(**nbr mod i == 0)

som← som+i

**FSI**

**fpour**

**SI(**nbr == som)

Afficher(nbr,«est un nombre parfait:  »)

**Sinon**

Afficher(nbr,«n’est un nombre parfait:  »)

**FSI**

**Fin**

Resolution Exercice 13

**Programme TestAnnee**

t**ype date = Structur**

**debut**

**jj : entier**

**mm : entier**

**an : entier**

**fin**

var d : date

**Debut**

Afficher(«Saisir jour , mois, annee :  »)

Saisir(s d ,jj , d,mm , d,an)

***SI***(d,jj ≥ 1 OU d,jj ≥ )

Alors

***SI***(d,mois ≥ 1 OU d,mois ≥ 12)

Alor

***SI***(d,an ≥ 1000 OU d,an ≥ 2000)

Afficher(«l’année est valide »)

**FSI**

**FSI**

**FSI**

Afficher(«l’année est invalide »)

**Fin**

**Ecrire 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 15 version 1

**Programme CalculSommeNombre**

var nombre,som,i : entier

**Debut**

Afficher(«Saisir un nombre  »)

Saisir(nombre)

som → 0

i→ 1

**Repeter**

som → som+i

i → i+1

**Jusqua**(i ≥ nombre)

Afficher(« **l**a somme des entiers jusqu'à ce nombre est :»,som)

**Fin**

Exercice 15 version 2

**Programme CalculMoyenneNombre**

var som\_element,i : entier

moyenne : reel

**Debut**

Afficher(«Saisir un nombre  »)

Saisir(nombre)

nbr\_element → 0

i→ 1

**Repeter**

som → som+i

i → i+1

**Jusqua**(i ≥ nombre)

moyenne → som/nombre

Afficher(« **l**a moyenne des entiers jusqu'à ce nombre est :»,moyenne)

**Fin**