**TP1**

**Exercice 1:**

1. **Analyse:**
2. **Entrés/Sorties:**

a

b x(x1,x2)

c

1. **Traitement:**

-Lecture de données(a,b,c).

-Calcul d.

1. **Algorithme:**

Algorithme Equation\_2éme\_degrée

Variable a,b,c,d,x1,x2 :réel.

Début

Ecrire (“Donner la valeur de a,b,c”).

Lire (a,b,c)

Si a = 0 et b ≠ 0 alors

x1 <-- -c/b

Ecrire (“la solution est” , x1)

Sinon

Si d = 0 alors

x1 <-- -b/2a

Ecrire (“la solution est” , x1)

Sinon

Si d > 0 alors

x1 <-- (-b + racine d)/2a

x2 <-- (-b - racine d)/2a

Ecrire (“x1 =”, x1)

Ecrire (“x2 =” ,x2)

Sinon

Ecrire (“x1 =(“ , -b , “+I” , racine(-d) , “)/” , 2a)

Ecrire (“x1 =(“ , -b , “-I” , racine(-d) , “)/” , 2a)

Finsi.

Finsi.

Finsi.

Fin.

1. **Algorithme ---- > Programme:**

#include <stdio.h>

#include<stdio.math>

Int main () {

//declaration de variable

float a,b,c,d,x1,x2 ;

//lecture de données

Printf (“Donnez la valeur de a,b,c”)

Scanf (“ %f %f %f ” &a &b &c) ;

If ( a == 0 ) & & ( b != 0 ) {

x1 = -(c/b)

x2 == x1

Printf ( “la solution est : %f “, &x1 ) ;

} else

d == b^2 - (4\*a\*c)

If ( d == 0 ) {

x1 == ( -b + racine d) / 2a

x2 == x1

Printf (“la solution est : %f, &x1)

} else

If (d > 0) {

x1 == ( -b + racine -d) / 2a

x2 == ( -b - racine -d) / 2a

Printf (“x1 =” %f, &x1)

Printf (“x2=” %f, &x2)

} Else

Printf (“x1 =(“ , -b, “+I”, racine -d, “)/” , 2a)

Printf (“x1 =(“ , -b, “-I”, racine -d, “)/” , 2a) }

}

**Exercice 2:**

1. x=3 y=3

x=3 y=2

x=3 y=-1 z=2

x=3 y=2 z=2

x=1 y=2 z=2

1. A: n=10 p=10 q=10 r=1

B: n=10 p=5 q=10

C: n=15 p=11 q=10

1. 12\_\_8

**Exercice 3:**

1. **a-**

* **analyse:**

Somme (s) des nombres pairs.

n

Produit (p) .

* **Traitement:**

-lecture de n.

- si n < 2 S <-- P <-- 0

Sinon S = S + 2

S = S + 4

,,

,,

S = S + 10

Valeur initiale : S=0 ,P=1, i =2

Introduction qui se répète : S=S+i

Condition d’arret : i>n

* **Programme :**

#include <stdio.h>

#include <math.h>

Int main ()

//declaration de variable

Int n,S,P,i ;

//initialisation

S=0

P=1

i=2

For ( i <= n) {

S+ = i ;

P+ = i ;

i = i +2 ;

}

Printf (“S= %d, P= %d”, &&)

}

**b-**

#include <stdio.h>

#include <math.h>

Int main ()

//declaration de variable

Int n,S,P,i ;

//initialisation

S=0

P=1

i=2

while ( i <= n) {

S+ = i ;

P+ = i ;

i = i +2 ;

}

Printf (“S= %d, P= %d”, &&)

}

**c-**

#include <stdio.h>

#include <math.h>

Int main ()

//declaration de variable

Int n,S,P,i ;

//initialisation

S=0

P=1

i=2

while ( i <= n) do

{ S+ = i ;

P+ = i ;

i = i +2 ;

}

Printf (“S= %d, P= %d”, &&)

}

**Exercice 4:**

Algorithme jeu mystère

**Variables**  
nombreMystère, saisieUtilisateur, nombreCoups: nombres  
**Début algorithme**  
nombreMystère <-- random(1;99)  
nombreCoups <-- 0  
saisieUtilisateur <-- -1  
 tant que nombreMystère ≠ saisieUtilisateur  
 Ecrire ("Veuillez entrer un nombre.")  
 Lire saisieUtilisateur  
 Si saisieUtilisateur = nombreMystère  
 Ecrire ("Vous avez gagné! ")  
 Sinon  
 Si saisieUtilisateur < nombreMystère  
 Ecrire ("Le nombre mystère est plus grand!")  
 Sinon  
 Ecrire ("Le nombre mystère est plus petit!")  
 Finsi.  
 Finsi.  
 nombreCoups <-- nombreCoups + 1  
 Fin tant que.  
Ecrire (nombreCoups, " coup(s) ont été nécessaire(s) pour réussir.")  
**Fin algorithme.**