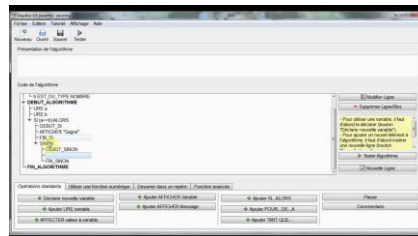
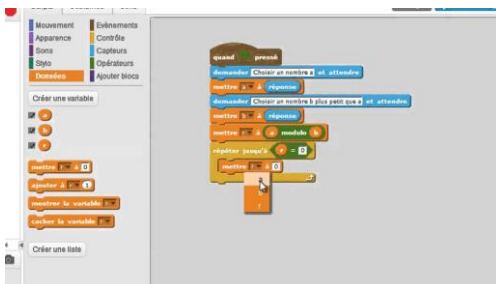


Cahier d'algorithmique – Programmation à l'aide de Scratch, Algobox, le langage C et Python



4^{ème} année

Table des matières

Partie 1 : Les fonctions.....	4
Activité 1 : Tableau de valeurs d'une fonction.....	4
Activité 2 : Détermination du maximum et du minimum d'une fonction.....	6
Partie 2 : Les Equations	8
Activité 3 : Résolution d'une équation par dichotomie.....	8
Activité 4 : Valeur approchée d'une racine carrée.	10
Activité 5 : Résolution d'une équation du second degré.	12
Partie 3 : Géométrie repérée.....	15
Activité 6 : Teste d'un quadrilatère.....	15

Notes importantes

A travers tous les exercices, les algorithmes Algobox seront transformés en Python (voir ce programme : Idle Python - File - New File) ou en C à l'aide de Notepad++ et de Code-Block. A toi à voir en fonction du programme de l'année en cours.

Vous allez donc créer ou compléter chaque algorithme à l'aide d'Algobox et ensuite le transformer après analyse en langage Python ou en C en fonction du projet développé courant de l'année.

Partie 1 : Les fonctions

Activité 1 : Tableau de valeurs d'une fonction

Objectif

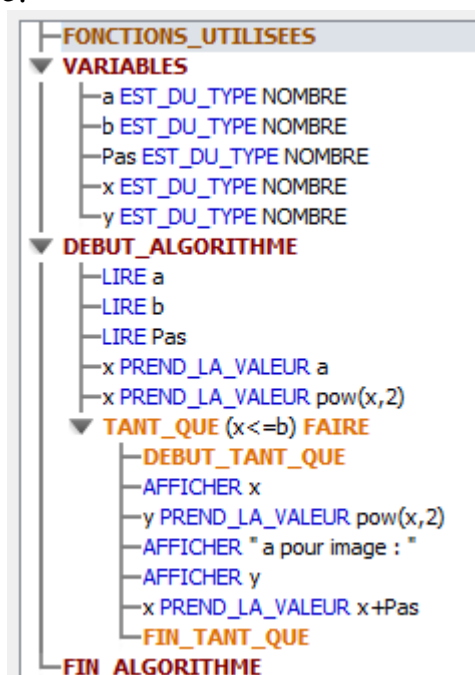
L'algorithme calcule les images d'une fonction pour les valeurs comprises dans l'intervalle $[a,b]$ avec un certain pas.

Ce que doit faire l'algorithme

Fonction carrée $x \rightarrow x^2$

A toi de jouer

Voici l'algorithme avec Algobox que tu vas devoir transformer en Python et en C.



a est le premier nombre qui va être mis au carré.

b sera la dernière valeur mise au carré.

Pas est la valeur de l'incrément.

x et y sont des variables utiles pour les calculs.

Par exemple, essaie l'algorithme avec les valeurs suivantes :

a = 1, b = 10 et Pas = 1

Tu auras le carré des nombres de 1 à 10.

Enregistre ton projet Algobox. Transforme-le en Python ou en C en fonction du programme de l'année.

La solution est à la page suivante.

Exercice en Python :

```
File Edit Format Run Options Window Help
a=float(input("Valeur de a ?"))
b=float(input("Valeur de b ?"))
pas= float(input("Valeur du pas ?"))
x=a
y=x**2
while x<=b:
    y=x**2
    print(x," a pour image ",y)
    x=x+pas
```

Exercice en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=0;
    int b=0;
    int pas=0;
    int x=0;
    int y=0;

    printf("Valeur de a ? ");
    scanf("%d", &a);
    printf("Valeur de b ? ");
    scanf("%d", &b);
    printf("Valeur du pas ? ");
    scanf("%d", &pas);

    x=a;
    y=x*x;
    while(x<=b)
    {
        y=x*x;
        printf("%d a pour image %d \n", x, y);
        x=x+pas;
    }

    system("pause");
    return 0;
}
```

Activité 2 : Détermination du maximum et du minimum d'une fonction

Objectif

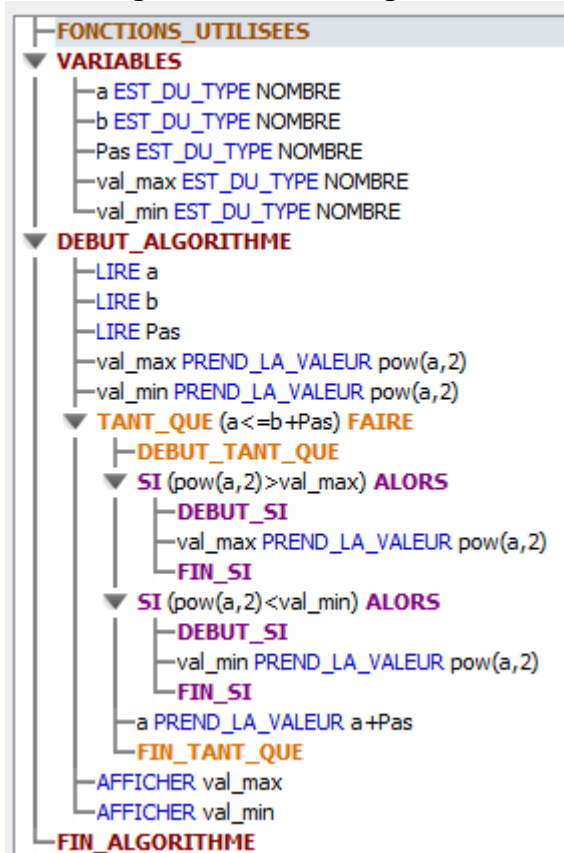
Créer une fonction qui détermine le max et le min d'une fonction.

Ce que doit faire l'algorithme

L'algorithme va déterminer le maximum et le minimum d'une fonction sur l'intervalle $[a,b]$

A toi de jouer

Voici l'algorithme avec Algobox.



Test ton algorithme à l'aide des données suivantes :

$A = -4$

$B = 3$

Pas = 0.1

Affichage 16 et 0. Cela veut dire que la fonction a pour maximum 16 et pour minimum 0 sur l'intervalle $[-4,3]$

Enregistre ton projet Algobox. Transforme-le en Python ou en C en fonction du programme de l'année.

Les solutions sont à la page suivante.

Exercice en Python :

```
Ex3.py - C:/Users/PortBart/Desktop/Cours d'algorithme
File Edit Format Run Options Window Help
a=float(input("Valeur de a ? "))
b=float(input("Valeur de b ? "))
Pas=float(input("Valeur de Pas ? "))
val_max=a**2
val_min=a**2
while a<b+Pas:
    if a**2>val_max:
        val_max=a**2
    if a**2<val_min:
        val_min=a**2
    a=a+Pas
print("Maximum : ",val_max)
print("Minimum : ",val_min)
```

Exercice en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=0;
    int b=0;
    int pas=0;
    int val_max=0;
    int val_min=0;

    printf("Valeur de a ? ");
    scanf("%d", &a);
    printf("Valeur de b ? ");
    scanf("%d", &b);
    printf("Valeur de pas ? ");
    scanf("%d", &pas);

    val_max=a*a;
    val_min=a*a;

    while(a<b+pas)
    {
        if(a*a>val_max)
        {
            val_max=a*a;
        }
        if(a*a<val_min)
        {
            val_min=a*a;
        }
        a=a+pas;
    }
    printf("Maximum : %d ", val_max);
    printf("Minimum : %d ", val_min);

    system("pause");
    return 0;
}
```

Partie 2 : Les Equations

Activité 3 : Résolution d'une équation par dichotomie.

Objectif

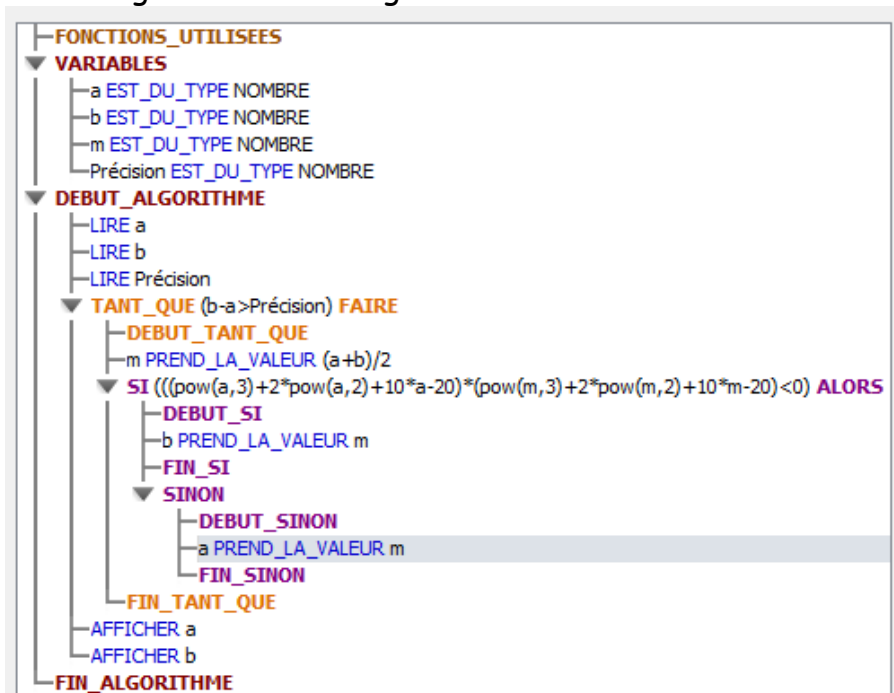
Résolution approchée d'une équation

Ce que doit faire l'algorithme

L'algorithme résout l'équation $f(x)=0$ de manière approchée par la méthode de la dichotomie en donnant un encadrement avec une certaine précision.

A toi de jouer

Voici l'algorithme avec Algobox.



Test ton algorithme à l'aide des données suivantes :

A = 1

B = 2

Précision = 0.0001

Affichage 1.36877...

et 1.36888...

Cela signifie que

l'équation

$x^3+2x^2+10x-20=0$ a

pour solution

approchée 1.3688

(au dix-millième près).

Enregistre ton projet Algobox. Transforme-le en Python ou en C en fonction du programme de l'année.

Les solutions sont à la page suivante.

Exercice en Python :

```
*Ex4.py - C:/Users/PortBart/Desktop/Cours d'algorithmique/Python/Ex4.py (3.5.2)*
File Edit Format Run Options Window Help
a=float(input("Valeur de a ? "))
b=float(input("Valeur de b ? "))
precision=float(input("Donnerz la précision : " ))
while b-a>precision:
    m=(a+b)/2
    if (a**3+2*a**2+10*a-20)*(m**3+2*m**2+10*m-20)<0:
        b=m
    else : a=m
print("La valeur approchée est comprise entre ",a," et ",b)
```

Exercice en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=0;
    int b=0;
    int precision=0;

    printf("Valeur de a ? ");
    scanf("%d", &a);
    printf("Valeur de b ? ");
    scanf("%d", &b);
    printf("Donnez la précision : ");
    scanf("%d", &precision);

    while(b-a>precision)
    {
        int m=(a+b)/2;
        if ((a*a*a+2*a*a+10*a-20)*(m*m*m+2*m*m+10*m-20)<0)
        {
            b=m;
        }
        else
        {
            a=m;
        }
    }
    printf("La valeur approchée est comprise entre %d et %d",a ,b);

    system("pause");
    return 0;
}
```

Activité 4 : Valeur approchée d'une racine carrée.

Objectif

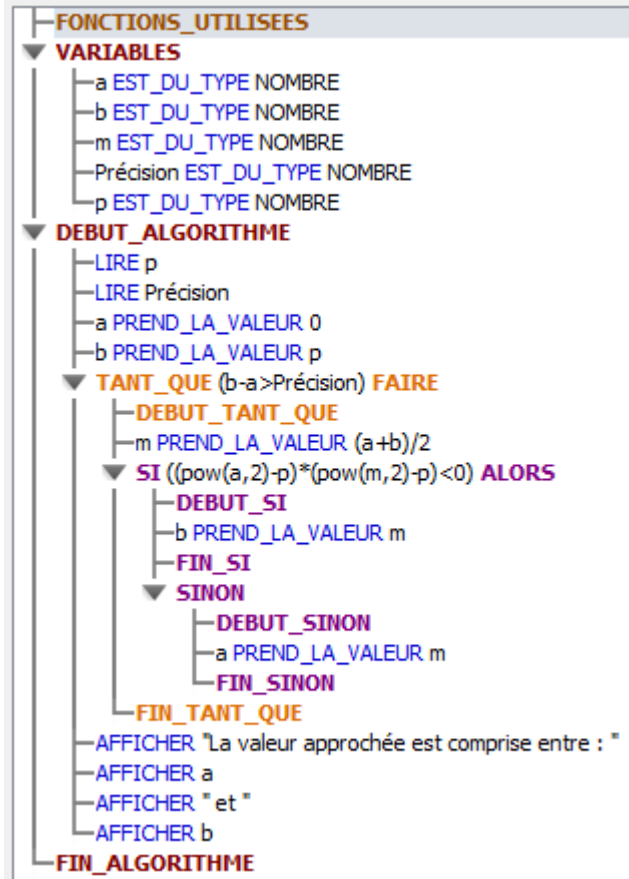
Algorithme qui donne la valeur approchée de n'importe quelle racine carrée.

Ce que doit faire l'algorithme

L'algorithme détermine une valeur approchée de n'importe quelle racine.

A toi de jouer

Voici l'algorithme avec Algobox.



Test ton algorithme à l'aide des données suivantes :

p = 2

Précision = 0.001

Précision = 0.0001

Affichage 1.414... et 1.415...

Enregistre ton projet Algobox. Transforme-le en Python ou en C en fonction du programme de l'année.

Les solutions sont à la page suivante.

Exercice en Python :

```
p=float(input("Nombre dont vous voulez la racine carrée ? "))
precision=float(input("Donne la précision : "))
a=0
b=p
while b-a>precision:
    m=(a+b)/2
    if (a**2-p)*(m**2-p)<0:
        b=m
    else: a=m
print("La valeur approchée est comprise entre ",a," et ",b)
```

Exercice en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int p=0;
    int precision=0;
    int a=0;
    int b=0;

    printf("Nombre dont vous voulez la racine carrée ? ");
    scanf("%d", &p);
    printf("Donnez la précision : ");
    scanf("%d", &precision);

    b=p;

    while(b-a>precision)
    {
        int m=(a+b)/2;
        if((a*a-p)*(m*m-p)<0)
        {
            b=m;
        }
        else
        {
            a=m;
        }
    }
    printf("La valeur approchée est comprise entre %d et %d",a ,b);

    system("pause");
    return 0;
}
```

Activité 5 : Résolution d'une équation du second degré.

Objectif

Résolution de l'équation $ax^2 + bx + c = 0$

Ce que doit faire l'algorithme

L'algorithme résout l'équation en utilisant la méthode du discriminant.

Petit rappel

Delta = $\text{pow}(b,2) - 4*a*c$ sachant que $\text{pow}(b,2) = b^2$

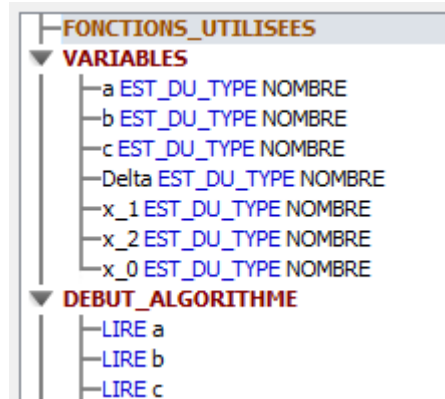
$x_1 = \frac{-b - \text{sqrt}(\text{Delta})}{2*a}$ sachant que sqrt = racine carrée

$x_2 = \frac{-b + \text{sqrt}(\text{Delta})}{2*a}$ sachant que sqrt = racine carrée

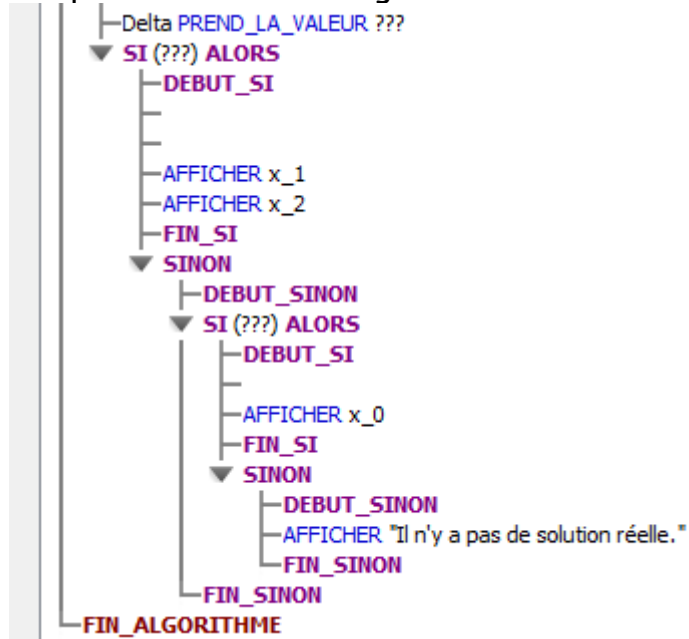
A toi de jouer

Voici l'algorithme avec Algobox en parti terminé.

On déclare les variables et on les lit :



Complète maintenant l'algorithme :



Test ton algorithme à l'aide des données suivantes :

A = 1

B = -5

C = 6

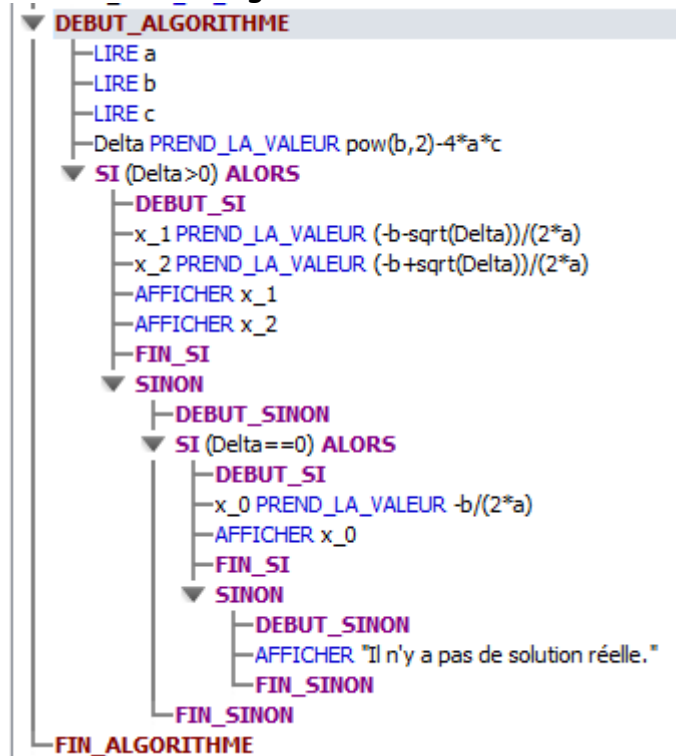
Tu devrais avoir comme solution $S=\{2;3\}$ pour cette équation.

Enregistre ton projet Algobox.

Transforme-le en Python ou en C en fonction du programme de l'année.

Les solutions sont à la page suivante.

Solution avec Algobox :



Algorithme en Python :

```
a=float(input("Valeur de a: "))
b=float(input("Valeur de b: "))
c=float(input("Valeur de c: "))
Delta=b**2-4*a*c
if Delta>0:
    x1=(-b-Delta**0.5)/(2*a)
    x2=(-b+Delta**0.5)/(2*a)
    print(x1)
    print(x2)
elif Delta==0:
    x0=-b/(2*a)
    print(x0)
else:
    print("Pas de solution")
```

Algorithme en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=0;
    int c=0;
    int b=0;
    int delta=b*b-4*c*a;

    printf("Valeur de a ? ");
    scanf("%d", &a);
    printf("Valeur de b ? ");
    scanf("%d", &b);
    printf("Valeur de c ? ");
    scanf("%d", &c);
    if(delta>0)
    {
        int x1=(-b-delta/2)/(2*a);
        int x2=(-b+delta/2)/(2*a);
        printf("%d", x1);
        printf("%d", x2);
    }
    else if (delta==0)
    {
        int x0=-b/(2*a);
        printf("%d", x0);
    }
    else
    {
        printf("pas de solution");
    }

    system("pause");
    return 0;
}
```

Partie 3 : Géométrie repérée

Activité 6 : Teste d'un quadrilatère.

Objectif

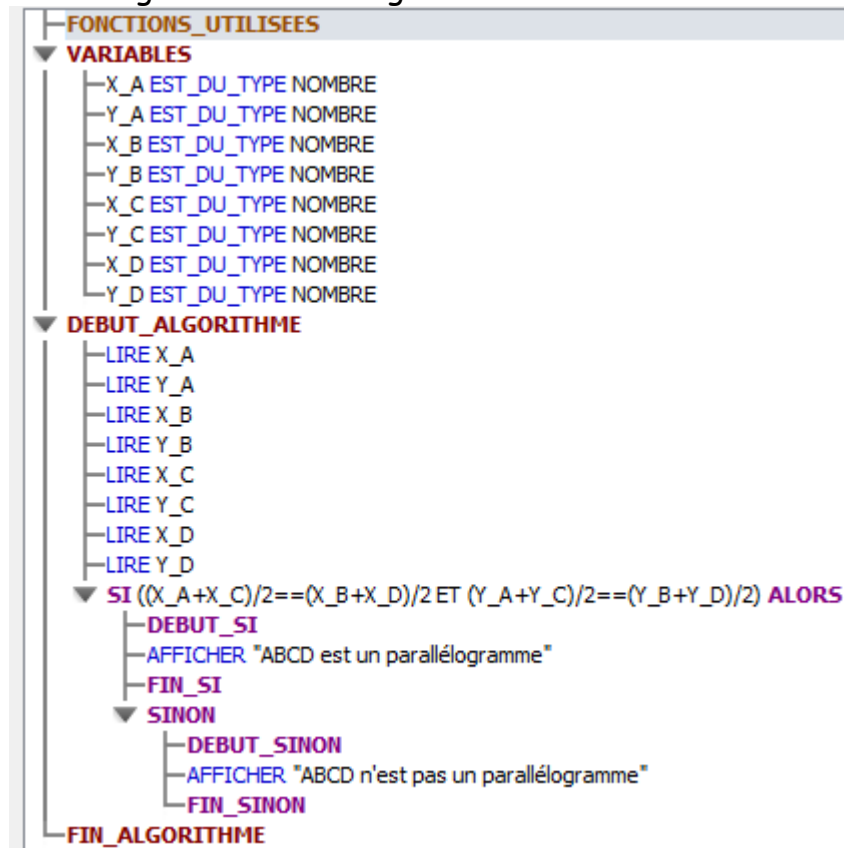
Algorithme qui teste si le quadrilatère ABCD est un parallélogramme.

Ce que doit faire l'algorithme

L'algorithme teste si les diagonales [AC] et [BD] du quadrilatère ABCD ont même milieu. Si c'est le cas, ABCD est un parallélogramme, sinon ce n'en est pas un.

A toi de jouer

Voici l'algorithme avec Algobox.



Test ton algorithme à l'aide des données suivantes :

$X_a = 2, Y_a = 1, X_b = 4, Y_b = 1, X_c = 5, Y_c = 2, X_d = 3, Y_d = 2$

Affichage : ABCD est un parallélogramme.

En Python comme en C, utilise les variables suivantes : $X_a, Y_a, X_b, Y_b, X_c, Y_c, X_d, Y_d$

Enregistre ton projet Algobox. Transforme-le en Python ou en C en fonction du programme de l'année.

Les solutions sont à la page suivante.

Exercice en Python :

```
Xa=float(input("Abcisse de A : "))
Ya=float(input("Ordonnée de A : "))
Xb=float(input("Abcisse de B : "))
Yb=float(input("Ordonnée de B : "))
Xc=float(input("Abcisse de C : "))
Yc=float(input("Ordonnée de C : "))
Xd=float(input("Abcisse de D : "))
Yd=float(input("Ordonnée de D : "))
if (Xa+Xc)/2==(Xb+Xd)/2 and (Ya+Yc)/2==(Yb+Yd)/2:
    print("ABCD est un parallélogramme")
else: print("ABCD n'est pas un parallélogramme")
```

Exercice en C :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int Xa=0;
    int Ya=0;
    int Xb=0;
    int Yb=0;
    int Xc=0;
    int Yc=0;
    int Xd=0;
    int Yd=0;

    printf("Absicce de A ? ");
    scanf("%d", &Xa);
    printf("Ordonnée de A ? ");
    scanf("%d", &Ya);
    printf("Absicce de B ? ");
    scanf("%d", &Xb);
    printf("Ordonnée de B ? ");
    scanf("%d", &Yb);
    printf("Absicce de C ? ");
    scanf("%d", &Xc);
    printf("Ordonnée de C ? ");
    scanf("%d", &Yc);
    printf("Absicce de D ? ");
    scanf("%d", &Xd);
    printf("Ordonnée de D ? ");
    scanf("%d", &Yd);
    if((Xa+Xc)/2==(Xb+Xd)/2 && (Ya+Yc)/2==(Yb+Yd)/2)
    {
        printf("ABCD est un parallélogramme");
    }
    else
    {
        printf("ABCD n'est pas un parallélogramme");
    }

    system("pause");
    return 0;
}
```