
Révisions - Informatique PSI

Exercice 1. Boucles/listes

Question 1.

```
1 def coefficients(A,B):
2     XA, YA = A
3     XB, YB = B
4     liste = []
5
6     if A[0] != B[0]:
7         liste.append((YB-YA)/(XB-XA))
8         liste.append(YA - a*XA)
9
10    return liste
```

Question 2.

```
1 if A[0] == B[0]:
2     print("L'équation de la droite est x =",A[0])
3 else:
4     a, b = coefficients(A, B)
5     print("L'équation de la droite est y =", a,"x +", b)
```

A=[1, 1],B=[2, 2] :L'équation de la droite est y = 1.0 x + 0.0

A=[2, 1],B=[2, 2] :L'équation de la droite est x = 2.0

Soit une liste existante nommée « points » contenant des points du plan sous forme de listes comme A et B. Ainsi, par exemple : points = [[2,4],[1,3],[0,7],[4,4]]

Question 3.

```
1 def nbr_pts(A, B, points):
2     N = 0
3     XA, YA = A
4     XB, YB = B
5     coeff = coefficients(A, B)
6     lenPts = len(points)
7
8     for i in range (lenPts):
9         XP, YP = points[i]
10
11    if XA == XB AND XP == XA:
12        N += 1
13    else:
14        a, b = coeff
15        if YP == a*XP+b:
16            N += 1
17
18    return N
```

Exercice 2. Dichotomie

Question 1.

```
1 def f(x):
2     Val = x**2 + x - 2
3     return Val
```

Question 2.

```

1 def new_int(f,Int):
2     x1,x2 = int
3     xm = (x1+x2)/2
4
5     if f(x1)*f(xm) <= 0:
6         x2 = xm
7     else:
8         x1 = xm
9     delta_x = abs(x2 - x1)
10    int_f = [x1,x2]
11    return int_f, delta_x

```

Question 3.

```

1 def dichotomie(f, int, crit_x): # Résolution d'équation par dichotomie
2     it = 0
3     x1, x2 = int
4     delta_x = x2 - x1 # Critère d'abscisses
5     while Delta_x > 2*Crit_x:
6         it += 1
7         int, delta_x = new_u=int(f, int)
8         res = (int[0] + int[1]) / 2
9     return res, delta_x, it

```

Question 4.

```

1 Crit = 1e-5
2 xs = - 1 / (2*1)
3 xc = xs - 10
4 int0 = [xc,xs]
5 sol, delta, it = dichotomie(f, int0, crit)
6 print("Solution: ",sol," - Delta: ",delta," - Itérations: ",it)
7
8 xc = xs + 10
9 int0 = [xs,xc]
10 sol, delta, it = dichotomie(f, int0, crit)
11 print("Solution: ",sol," - Delta: ",delta," - Itérations: ",it)

```

Exercice 3. Euler

Question 1. e^x

Question 2.

$$\begin{cases} y_0 = z(t_0) \\ y_{k+1} = y_k + hF(t_k; y_k), \quad 0 \leq k \leq N-1 \end{cases}$$

Question 3.

```

1 def euler(F, a, b, y0, h):
2     t = 0
3     y = y0
4     X = [t]
5     Y = [y]
6     X, Y = [0], [1]
7
8     for i in range(a, b, h):
9         y = y + h*F(t, y)
10        t = t + h
11        X.append(t)
12        Y.append(y)
13
14    return X, Y

```

Question 4.

```

1 import matplotlib . pyplot as plt
2 import numpy as np
3
4 F = lambda t, y : y
5 X,Y= euler (F, 0, 3, 1, 0.1)
6 plt. plot(X,Y)
7
8 Y1 = []
9 for x in X:
10     Y1.append(np.exp(x))
11
12 plt. plot(X,Y1)
13 plt. show ()

```

Exercice 4. Intégration numérique

Question 1.

```

1 N = len(V)
2 T = []
3 for i in range(N):
4     T.append(i*0.1)
5 print(T)

```

Question 2.

```

1 v_ms = []
2 for i in range(N):
3     v_ms.append(V[i]*1000/3600)

```

Question 3.

```

1 def f_integre(temps, liste):
2     taille = len(temps)
3     integrale = 0
4     for i in range(taille-1):
5         dT = temps[i+1] - temps[i]
6         val = (liste[i]+liste[i+1])/2
7         ajout = val * dT
8         integrale += ajout
9     return integrale

```

Question 4.

```

1 d = f_integre_trapezes(t, v_ms)
2 d = round(d,1)
3 print("Distance parcourue: ",d," m")

```

Exercice 5. Salles de cours

Question 1. SELECT DISTINCT age FROM eleves ;

Question 2. SELECT nom, prenom FROM eleves WHERE classe = "MPSI" ;

Question 3. SELECT DISTINCT age FROM eleves WHERE classe = "MPSI" OR classe = "PCSI" ;

Question 4. SELECT DISTINCT ville FROM Eleves ;

Question 5. SELECT nom, prenom FROM eleves WHERE age BETWEEN 18 AND 130 ;

Question 6. SELECT eleves.prenom, eleves.nom, profs.salle FROM eleves JOIN profs ON eleves.classe = profs.classe ;

Question 7. SELECT DISTINCT profs.nom FROM eleves JOIN profs ON eleves.classe = profs.classe WHERE eleves.age < 18 ;

Question 8. SELECT COUNT(*) FROM eleves WHERE ville = "Paris" ;

Question 9. SELECT MIN(age), MIN(age) FROM eleves WHERE classe = "MP" OR classe = "PC" OR classe = "PSI" ;

Question 10. SELECT AVG(age) FROM eleves.