

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Altan

Prénom : Dilara

Classe : **1M12**

<b>Nombre de points</b>	<b>3 / 11</b>
<b>Note</b>	<b>2.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
0,5	0,5	0	0	0	0

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0,5	0,5	0	0	0	0

```
1 nombre = int(input (" li : "))
2 nops = 0
3 s = 0
4 if i in range [1]
5     if i est plus petit que li
6         print(i, "s= s+(1/i*(i+1))")
7         i=i+1
8         nops=nops+1
9         print("resultat : ",res)
10    if i est plus grand que li
11        print(i, "output(s,nops)"
```

Ligne 1 : la variable d'entrée de l'algorithme est li

Ligne 4 : ce n'est pas un test mais une boucle for (ou while). La fonction range() demande comme argument (entre parenthèses) la borne supérieure (ici li)

Ligne 5 et 10 : il n'y a pas de test dans l'algorithme. D'autre part, le test conditionnel se fait en utilisant un opérateur de comparaison (<, >, <=, >=, ==, !=)

Ligne 6:il n'y a aucune sortie dans le bloc d'instructions de la boucle

Ligne 11 : cette sortie doit s'effectuer en dehors de la boucle, soit au niveau du programme principal. L'argument de la fonction print() est composé des valeurs des variables s et nops pas d'une chaîne de caractères.

## Respect consigne (1 pt)

1pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Berisha

Prénom : Auron

Classe : **1M12**

<b>Nombre de points</b>	<b>3.3 / 11</b>
<b>Note</b>	<b>2.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0	0	0,3		0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0	0	0	0	0

```

1 int(input("nt:"))
2 signe=1
3 nops=0
4 estim=0
5
6 i in range (nt=signe):
7     if (i<nt*2)
8         print("estim=estim+(signe*(4/i.:))
9         else:
10            signe = signe*-1:
11            elif nops=nops+1:
12                else:
13                    int(output("estim,nops":))
14

```

Ligne 1 : la sortie de la fonction input() doit être affectée à la variable nt

Ligne 6 : une boucle s'écrit avec le mot-clef python for. La fonction range() prend comme argument un entier (ici nt)

Ligne 7 + 9 + 11 + 12 (!) : il n'y a pas de test dans l'algorithme

Ligne 8 : c'est une instruction dans l'algorithme, pas une sortie (print())

Ligne 13 : une sortie s'écrit avec la fonction print() et non output. L'argument de la fonction devrait être la valeur des variables estim et nops (voir algorithme). Pas une chaîne de caractères.

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Blatti

Prénom : Guillaume

Classe : **1M12**

<b>Nombre de points</b>	<b>8.8 / 11</b>
<b>Note</b>	<b>5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	0,5		1

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,5	0,3	1	0	0

```

1 nt = int(input('entrez votre nombre'))
2 signe = 1
3 nops = 0
4 estim = 0
5 i = 1
6 for i in range(nt):
7     if i < nt*2:
8         print(estim,nops)
9     else:
10        estim= estim+(signe*(4/i))
11        signe = signe*-1
12        nops = nops+1
13        i = i+2
14

```

Ligne 6 : (voir algorithme). La boucle s'écrit for i in range(1,2\*nt,2)

Ligne 7 + 9 : il n'y a pas de test dans l'algorithme

Ligne 8 : la sortie se fait en dehors de la boucle for

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Crivello

Prénom : Charlotte

Classe : **1M12**

<b>Nombre de points</b>	<b>8.5 / 11</b>
<b>Note</b>	<b>5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0	1	1		1

Voir feuille

**Exercice 2 : Algorithme vers programme (6 pts)**

variables	I/O	boucle	Prog OK	Question	Qs
1	1	0,5	1	0	0

```
1 nt=input("entrez un nombre:")
2 nt=int(nt)
3 nops=0
4 x=5
5
6 for i in range(x):
7     if i<nt:
8         x=0.5*(x+5/x)
9         nops=nops+1
10    else:
11        print(x,nops)
```

Ligne 6 : (voir algorithme) la borne supérieure de la boucle est la variable nt

Ligne 7 + 10 : il n'y a pas de test dans cet algorithme

Ligne 11 : la sortie se fait en dehors de la boucle

**Respect consigne (1 pt)**

*1pt*



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Ebrahimi

Prénom : Shamila

Classe : **1M12**

<b>Nombre de points</b>	<b>4 / 11</b>
<b>Note</b>	<b>3</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
0	1	0	1	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0,5	0,5	0	0	0	0

```

1 nops
2 s
3 i
4 li = int(input(li))
5 print ("s","nops")
6 for li in range (s,nops):
7     if i<li:
8         print ("s","nops")
9     else:
10        s=s+(1/2**i)
11        i=i+1
12        nops=nops+1
13    print ("s","nops")

```

Ligne 1, 2 et 3 : affectation des variables nops, s et i (nops = 0, s = 0, i = 0)

Ligne 4 : L'argument de la fonction input() est une chaîne de caractères (pas li)

Ligne 5 : n'existe pas dans l'algorithme

Ligne 6 : l'itérateur de la boucle est la variable i, la borne supérieure est li (for i in range(li) )

Ligne 7 : test inexistant dans l'algorithme. Attention, dans un test if ... else ... le bloc d'instructions doit être avec une indentation de 4 espaces

Ligne 13 : l'output se trouve en dehors de la boucle. Il faut afficher la valeur des variables nops et s (et pas les chaînes de caractères «nops» et «s»)..

Correction :

```

li = int(input("Borne sup : "))
nops = 0
s = 0
for i in range(li):
    s = s + (1/2 ** i)
    nops = nops + 1
print(s,nops)

```

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Grandjean

Prénom : Léa

Classe : **1M12**

<b>Nombre de points</b>	<b>/ 11</b>
<b>Note</b>	

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>

**Exercice 2 : Algorithme vers programme (6 pts)**

variables	I/O	boucle	Prog OK	Question	Qs

```
1 For i in range():  
2
```

**Respect consigne (1 pt)**

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Heng

Prénom : Lily

Classe : 1M12

<b>Nombre de points</b>	<b>6 / 12</b>
<b>Note</b>	<b>3.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0,5	1	1	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0.5	0.5	0.5	0	0	0

```

1 N = int(input("Entrez votre taille" ))
2 nops = 0
3 estim = 0
4 i = 1
5 while i < taille + 1:
6 for taille in range(taille+1) :
7     if(i < taille + 1 ) :
8         print(taille,"est estim,nops".)
9     elif:
10         (estim = 1 + 1/estim)
11     elif :
12         (i = i + 1)
13     else :
14         (nops = nops + 1)

```

Ligne 1 : le nom de la variable est taille (pas N)

Ligne 3 : la variable estim est initialisée à 1

Ligne 5 et 6 : il faut choisir une boucle for ou une boucle while, mais pas les deux

Ligne 7 + 9 + 11 + 13 : il n'y a aucun test dans cet algorithme

Ligne 10 + 12 + 14 : les parenthèses sont inutiles dans une instruction

Ligne 8 : l'output se trouve à l'extérieur de la boucle et doit afficher les valeurs des variables estim et nops

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Kone

Prénom : Mélissa

Classe : **1M12**

<b>Nombre de points</b>	<b>/ 11</b>
<b>Note</b>	

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>

**Exercice 2 : Algorithme vers programme (6 pts)**

variables	I/O	boucle	Prog OK	Question	Qs

```
1 For i in range():  
2
```

**Respect consigne (1 pt)**



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Lacotte

Prénom : Camille

Classe : **1M12**

<b>Nombre de points</b>	<b>6.5 / 12</b>
<b>Note</b>	<b>3.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0	1	1	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,5	1	0	0	0

```
1 nt = int(input("entrez la valeur de nt"))
2 nops = 0
3 x = 2
4
5 i = 1
6
7 while i < nt:
8     if i < nt:
9
10 else:
11     if x = 0,5*(x+5/x):
12         i = i + 1
13         nops = nops + 1
14
```

Ligne 8 + 9 : il n'y a pas de test dans l'algorithme

Ligne 11 : un nombre en virgule flottante utilise le point et non la virgule (0.5 et non 0,5). De plus il n'y a pas de test dans l'algorithme

Manque la sortie en sortie de boucle

## Respect consigne (1 pt)

*1pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Lucrezia

Prénom : Davide

Classe : **1M12**

<b>Nombre de points</b>	<b>0 / 11</b>
<b>Note</b>	

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>

Exercice 2 : Algorigramme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs

```
1 For i in range() :
2
```

Respect consigne (1 pt)

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Maliutina

Prénom : Arina

Classe : **1M12**

<b>Nombre de points</b>	<b>12 / 11</b>
<b>Note</b>	<b>6</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1		1

Voir feuille

**Exercice 2 : Algorithme vers programme (6 pts)**

variables	I/O	boucle	Prog OK	Question	Qs
1	1	1	1	1	1

```
1 nt = int(input("entrez la valeur de nt: "))
2
3 nops = 0
4 x = 5
5 i = 1
6
7 while i < nt:
8     x = 0.5 * (x + 5 / x)
9     i += 1
10    nops += 1
11
12 print("x = ", x)
13 print("nops = ", nops)
```

**Respect consigne (1 pt)***1pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Maurer

Prénom : Olivier

Classe : **1M12**

<b>Nombre de points</b>	<b>8.5 / 11</b>
<b>Note</b>	<b>5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1		1

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,5	1	0	0	0

```

1 nt = float(input("entrez un nombre : "))
2
3 signe = 1
4 nops = 0
5 estim = 0
6 i = 1
7
8 while i < nt * 2 :
9     if i < nt * 2 :
10         estim = estim + (signe * (4/i))
11         signe = signe * -1
12         nops = nops + 1
13         i = i+1
14     else:
15         print("estim, nops")

```

Ligne 1 : le casting vers un flottant est incorrect puisque la condition se fait ensuite avec un entier (voir ligne 8)

Ligne 9 + 14 : il n'y a pas de test dans l'algorithme

Ligne 15 : doit se trouver à l'extérieur de la boucle (voir algorithme). D'autre part, on veut afficher les valeurs des deux variables estim et nops, pas la chaîne de caractères «estim, nops»

## Respect consigne (1 pt)

*1pt*



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Nerny

Prénom : Alessia

Classe : **1M12**

<b>Nombre de points</b>	<b>8.5 / 11</b>
<b>Note</b>	<b>5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1		0,5

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	1	1	0	0	0

```
1 nt = int(input("nt:"))
2
3 nops = (0)
4 x = (5)
5 i = (1)
6
7
8 while i<nt :
9     print(x, nops)
10 else:
11     nops=nops+1
12     i=i+1
13     x=0.5*(x+5/x)
```

Ligne 10 : que fait ce else ? Il n'y a pas de test dans l'algorithme. Sans cette ligne votre code est parfaitement correct

Ligne 9 : à l'extérieur de la boucle

## Respect consigne (1 pt)

*1pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Origas

Prénom : Mathilda

Classe : **1M12**

<b>Nombre de points</b>	<b>4.75 / 12</b>
<b>Note</b>	<b>3</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
0,5	0	0,5	0	0,5	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,5	0,75	0	0	0

```

1 li = input("entrez un nombre:")
2 li = int(li)
3 nops = 0
4 s = 0
5 i = 0
6 while i < len(li):
7     if li[i] == 0:
8         print("li ,nops et s sont tous egale a 0")
9     else:
10        if s == s + (1/2**i):
11            li.append("inferieur a 0")
12        else:
13            nops = nops + 1
14        i = i + 1
15 print("nombre finale")

```

Ligne 6 : la borne inférieure de la boucle while est la variable li (et non une liste li dont la longueur est donnée avec la fonction len()).

Ligne 7 et 9 : il n'y a pas de test dans cet algorithme. Attention cependant à l'indentation du bloc d'instructions else (au même niveau vertical que if).

Ligne 11 : li n'est pas une liste (voir lignes 1 et 2)

Ligne 15 : la sortie de l'algorithme est l'affichage des valeurs des variables s et nops.

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Perche

Prénom : Lorick

Classe : **1M12**

<b>Nombre de points</b>	<b>8 / 12</b>
<b>Note</b>	<b>3.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0	0	0	0,5	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,75	0,75	1	0	0

```
1 taille=int(input("Entrez la taille:"))
2 nops=0
3 estim=1
4 i=1
5 for i in range (nops,taille,1):
6     if (i < taille+1):
7         estim=1+1/estim
8         print("tu feras",taille,"cm à",estim)
9
10    else:
11        print("tu feras",taille,"cm à",estim)
```

Ligne 6 + 10 : il n'y a pas de test dans cet algorithme

Ligne 5 : la boucle commence par 1 (valeur de l'itérateur i = 1)

Ligne 8 + 11 : la sortie (voir algorithme) doit afficher estim et nops en dehors de la boucle

Il manque le calcul de nops dans le bloc d'instructions de la boucle

## Respect consigne (1 pt)

*1 pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Poretti

Prénom : Andrej

Classe : **1M12**

<b>Nombre de points</b>	<b>5 / 11</b>
<b>Note</b>	<b>3.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0	1	0,5		0.5

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0	0	1	0	0	0

```
1 signe = [ 1.0 ]
2 nops = [ 0 ]
3 estim = [ 0 ]
4 i = 1
5 nt = []
6
7 while( i<nt*2 ):
8     if ( estim == estim + (signe * (4/i))):
9         (signe ==signe * -1)
10        (nops == nops + 1)
11 else :
12     print( estim, nops)
```

Ligne 1, 2, 3 et 5 : ce sont des entiers, pas des listes

Lignes 8 et 11 : il n'y a pas de test dans cet algorithme

Lignes 9 et 10 (ainsi que votre exercice 1) : l'opérateur d'affectation est «= $\Rightarrow$ » alors que celui de comparaison (égalité) est «==»

Ligne 12 : la sortie doit se trouver à l'extérieur de la boucle

Il manque une entrée pour la variable nt

## Respect consigne (1 pt)

1 pt



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Rejaibi

Prénom : Inès

Classe : **1M12**

<b>Nombre de points</b>	<b>3.5 / 12</b>
<b>Note</b>	<b>2.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0,5	0	0	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0,5	0,5	0	0	0	0

```

1 base=int(input(nt))
2 nops=0
3 X=2
4 i=1
5 'if i < nt'
6   for i in range(x=0.5*(x+5/x)):
7       else:
8       for i in range(i=1)
9       nops = nops + 1
10      print(x,nops)

```

Ligne 1 : le nom de la variable d'entrée est nt (pas base qui provient de l'exercice 1). L'argument de la fonction input() doit être absolument de type str (chaîne de caractères)

Ligne 2 : vous déclarez x en majuscule (puis plus loin en minuscule), la casse de la police (majuscule ou minuscule) est importante dans un code informatique

Ligne 5 : Une chaîne de caractères posée au milieu du code a peu de chances de faire fonctionner celui-ci

Ligne 6 : indentation mauvaise. La signature de la fonction range() est range(start, stop, step) avec step non compris. Ce que vous avez écrit est une instruction faisant partie du bloc d'instructions de la boucle.

Ligne 7 : que fait un else tout seul orphelin sans if ? Il n'y a pas de test dans cet algorithme

Ligne 8 : pourquoi une deuxième boucle (voir algorithme)

Ligne 10 : la sortie se trouve à l'extérieur de la boucle

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Rossi

Prénom : Sofia

Classe : **1M12**

<b>Nombre de points</b>	<b>5.75 / 12</b>
<b>Note</b>	<b>3.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
0,25	0,5	0	0	0	0

```

1 nt = input("Entrez nt:")
2 nops = [0]
3 x = [2]
4 i = 1
5
6
7 if (i < str(nt)):
8     for i in range((x == 0.5*(x + 5/x))):
9         i.append(i = 1 + 1)
10        nops.append(nops = nops + 1)
11 else :
12     print(x, nops)

```

Ligne 2 et 3 : les variables nops et x ne sont pas des listes (mais des entiers)

Ligne 7 (et 11) : il s'agit d'une boucle, pas d'un test simple (voir la flèche qui revient dans l'algorithme)

Ligne 7 : condition de sortie du test incorrect. On compare des variables de même types. La variable i est de type entier, nt doit aussi être entier et non chaîne de caractères)

Ligne 8 : La signature de la fonction range() est range(start, stop, step) avec step non compris. Ce que vous avez écrit est une instruction faisant partie du bloc d'instructions de la boucle.

Ligne 9 et 10 : si les variables i et nops étaient des listes (ce qui n'est pas le cas, ce sont des entiers), alors l'argument de la fonction append() est une valeur ou une variable, pas une instruction

Ligne 12 : la sortie doit être en dehors de la boucle

## Respect consigne (1 pt)

1 pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Roux

Prénom : Nicolay

Classe : **1M12**

<b>Nombre de points</b>	<b>5.25 / 12</b>
<b>Note</b>	<b>3</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0,5	0	1		0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,5	0,25	0	0	0

```

1 nt = int(input("Entrez un nombre : "))
2 nops = 0
3 x = 2
4 i = 1
5
6 if i > 0:
7     for y in range(-nt):
8         x = 0.5 * (x + 5/x)
9         i = i + 1
10        nops = nops + 1
11
12 else :
13     for y in range(nt):
14         if i < nt
15     print(nt, " est égale à")

```

Ligne 6 et 12 : il n'y a pas de test dans cet algorithme. Il y a une boucle qui commence à 1 et fini à (non compris) nt. D'autre part, une condition de test s'écrit avec les opérateurs de comparaison vus au cours (>, <, >=, <=, ==, !=)

Ligne 8, 9, 10 : il s'agit du bloc d'instruction de la boucle précédemment explicitée (for i in range(nt)). Attention à l'indentation de la ligne 10

Ligne 14 : d'une part il n'y a pas de test dans cet algorithme, mais s'il y en avait un, sa syntaxe se fait en n'oubliant pas les deux points finaux afin d'explicitier le bloc d'instructions du test.

Ligne 15 : la sortie fait partie du programme principal et n'est pas compris dans le test (la boucle). En sus, il faut afficher x ainsi que nops et non nt (qui est la valeur de la variable entrée par l'utilisateur)

## Respect consigne (1 pt)

1pt

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Schertenleib

Prénom : Lucie

Classe : **1M12**

<b>Nombre de points</b>	<b>/ 11</b>
<b>Note</b>	

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>

Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs

```
1 For i in range() :
2
```

Respect consigne (1 pt)



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Suter

Prénom : Yann

Classe : **1M12**

<b>Nombre de points</b>	<b>7.75 / 11</b>
<b>Note</b>	<b>4.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1		1

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,75	0	0	0	0

```

1 nt = float(input("Entrez un nombre: "))
2
3 nops = 0
4 x = 5
5 i = 1
6
7 if i < nt:
8     print(" Votre nombre est plus grand que 1",)
9 else :
10    nt = float(input("Entrez un autre nombre: "))
11
12    x = float(0.5*(x+5/x))
13    i = int(i+1)
14    nops = int(nops+1)
15    if i < nt and x and nops:
16        print("votre nombre est plus grand que",x,nops)

```

Ligne 1 : nt est un entier (voir ligne 7) pas un flottant

Ligne 7 + 9 + 15 : il n'y a pas de test dans cet algorithme mais une boucle

Ligne 8 + 9 : ces sorties (print()) ne sont pas dans l'algorithme. Il ne faut pas inventer un programme, il faut suivre l'algorithme représenté par le formalisme algorithme

Ligne 14 : il est inutile de caster en entier un nombre qui est déjà entier (nops + 1).

Ligne 16 : la sortie se trouve dans le programme principal (aligné avec la marge de gauche)

## Respect consigne (1 pt)

*1pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Tarabori

Prénom : Margaux

Classe : **1M12**

<b>Nombre de points</b>	<b>6.5 / 11</b>
<b>Note</b>	<b>4</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	0,5	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	1	0	0	0	0

```
1 taille = int(input("Entrez votre taille :"))
2 nops = 0
3 estim = 1
4 i = 1
5 taille = []
6
7 for taille in range(i):
8     if (i < taille + 1):
9         taille.append = i + 1
10    estim = 1 + 1 float/(estim)
11
12 print(estim, nops)
```

Ligne 5 : inutile (taille n'est pas une liste puisque vous l'avez initialisée en ligne 1 correctement)

Ligne 7 : l'itérateur de la boucle est i, la borne supérieure est taille

Ligne 8 : il n'y a pas de test dans cet algorithme

Ligne 9 : voir remarque ligne 7 (taille n'est pas une liste)

Ligne 11 : il manque nops = nops + 1

## Respect consigne (1 pt)

*1pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Tinel

Prénom : Chloé

Classe : **1M12**

<b>Nombre de points</b>	<b>10.5 / 11</b>
<b>Note</b>	<b>6</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1	0	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	1	1	1	0,5	1

```
1 taille = int(input("Entrez une taille : "))
2 nops = 0
3 estim = 1
4 i = 1
5
6 while i < taille + 1:
7     estim = 1 + 1/estim
8     i = i + 1
9     nops = nops + 1
10
11 print("valeur de estim : ", estim)
12 print("nombre d'opération : ", nops)
```

Ligne 12 : l'algorithme contient qu'une seule sortie (estim + nops)

## Respect consigne (1 pt)

*1 pt*

# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Yilmaz

Prénom : Berfin

Classe : **1M12**

<b>Nombre de points</b>	<b>7.25 / 12</b>
<b>Note</b>	<b>4</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	1	1	1	0,25	0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	0,75	0,25	0	0	0

```
1 taille = input ("Entrez la taille:")
2 nops = 0
3 estim = 1
4 i = 1
5 for i in range(i < taille + 1):
6
7     (estim = 1+1/estim)
8     ( i = i+1)
9     ( nops = nops+1)
10 else :
11     print(estim,nops)
```

Ligne 1 : taille doit être de type entier (c'est la borne supérieure de la condition de sortie de la boucle)

Ligne 5 : la fonction range prend comme argument la borne supérieure : donc taille. Le test de sortie de la boucle est implicitement compris dans la syntaxe for i in range(taille) :

Ligne 7, 8, 9 : parenthèses inutiles et fausses. Attention à l'indentation (4 caractères espace depuis la marge de gauche)

Ligne 10 : il n'y a pas de test ni de else dans cet algorithme. Il y a une condition de sortie d'une boucle.

Ligne 11 : la sortie doit se faire à l'extérieur de la boucle et donc doit faire partie du programme principal (aligné sur le for)

## Respect consigne (1 pt)

1 pt



# Programmation

## TP noté – 2024-2025 (Correction)

Nom : Zaichko

Prénom : Diana

Classe : **1M12**

<b>Nombre de points</b>	<b>10.25 / 11</b>
<b>Note</b>	<b>5.5</b>

### Exercice 1 : Programme vers algorithme (6 pts)

<b>variables</b>	<b>entrées</b>	<b>sorties</b>	<b>tests</b>	<b>boucles</b>	<b>Algo OK</b>
1	0,5	1	0,75		0

Voir feuille

## Exercice 2 : Algorithme vers programme (6 pts)

variables	I/O	boucle	Prog OK	Question	Qs
1	1	1	1	1	1

```
1 nt=int(input("Entrez la valeur de nt:"))
2 signe=1
3 nops=0
4 estim=0
5 i=1
6 while i < nt*2:
7     nops+=1
8     signe*=-1
9     estim+=signe*(4/i)
10    i+=2
11 print("Estim:",estim,)
12 print("Nombre des operations:",nops)
```

Votre code produit le résultat -3.14 au lieu de 3.14. Cela est dû à l'inversion des lignes 8 et 9. Ou alors affectez la valeur -1 à signe en ligne 2

## Respect consigne (1 pt)

*1pt*