

France – iOi

Table des matières

NIVEAU 1.....	3
N1_C1. Affichage de texte, suite d'instructions.....	3
N1_C2. Répétitions d'instructions.....	4
N1_C3. Calculs et découverte des variables.....	7
N1_C4. Lecture de l'entrée.....	10
N1_C5. Tests et conditions.....	13
N1_C6. Structures avancées.....	14
N1_C7. Conditions avancées, opérateurs booléens.....	15
N1_C8. Répétitions conditionnées.....	15
NIVEAU 2.....	16
N2_C1. Nombres à virgules et autres outils.....	16
N2_C2. Découverte des tableaux.....	17
N2_C3. Chaînes de caractères.....	17
N2_C4. Fonctions.....	18
N2_C5. Programmer sur son ordinateur.....	19
NIVEAU 3.....	19
N3_C1.....	19
N3_C2.....	19
N3_C3.....	19
N3_C4.....	20
N3_C5.....	20
N3_C6.....	20
N3_C7.....	20
N3_C8.....	20

N3_C9.....	21
N3_C10.....	21
N3_C11.....	21
NIVEAU 4.....	21
N4_C1.....	21
N4_C2.....	22
N4_C3.....	22
N4_C4.....	22
N4_C5.....	22
N4_C6.....	22
N4_C7.....	23
N4_C8.....	23
N4_C9.....	23
NIVEAU 5.....	23
N5_C1.....	23
N5_C2.....	24
N5_C3.....	24
N5_C4.....	24
N5_C5.....	24
N5_C6.....	24
N5_C7.....	25
N5_C8.....	25
N5_C9.....	25
NIVEAU 6.....	25
N6_C1.....	25
N6_C2.....	26
N6_C3.....	26
N6_C4.....	26
N6_C5.....	26
N6_C6.....	26
N6_C7.....	27

NIVEAU 1

N1_C1. Affichage de texte, suite d'instructions

1) Hello world! :

```
print("Hello world!")
```

```
Hello world!
```

2) Présentation :

```
print("Coucou !")
print("Je m'appelle Camthalion")
print("Ma devise est 'Parler peu mais parler bien'.")
```

```
Coucou !
Je m'appelle Camthalion
Ma devise est 'Parler peu mais parler bien'.
```

Afficher du texte : erreurs possibles :

ATTENTION AUX PARENTHESES !

```
print "Bonjour")
↳ SyntaxError: invalid syntax
```

```
print("Bonjour")
↳ SyntaxError: unexpected EOF while parsing
```

ATTENTION AUX GUILLEMENTS !

```
print(Bonjour)
↳ NameError: name 'Bonjour' is not defined
```

```
print("Bonjour)
↳ SyntaxError: EOL while scanning string literal
```

```
print(Bonjour tout le monde)
↳ SyntaxError: invalid syntax
```

3) Plan de la montagne :

```
print("Tout droit tu grimperas,")
print("La clé tu trouveras,")
print("Habile tu seras,")
print("Quand tu les porteras,")
print("Et avec le chef tu reviendras !")
```

```
Tout droit tu grimperas,
La clé tu trouveras,
Habile tu seras,
Quand tu les porteras,
Et avec le chef tu reviendras !
```

4) Dans le fourré :

```
from robot import *
haut()
haut()
haut()
droite()
droite()
bas()
bas()
droite()
```

```
["startSimu"],
["haut"],
["haut"],
["haut"],
["droite"],
["droite"],
["bas"],
["bas"],
["droite"],
["displayMsg", "Le robot a bien trouvé la clé"]
```

Mouvements : nécessité du module :

Si on oublie la ligne qui fournit les instructions pour bouger le robot, comme dans le programme suivant :

```
haut()
droite()
bas()
gauche()
```

on obtient une erreur :

↳ line 1, in

```
haut()
```

NameError: name 'haut' is not defined

5) Empilement de cylindres :

```
from robot import *
deplacer(1, 2)
deplacer(1, 3)
deplacer(2, 3)
deplacer(1, 2)
deplacer(3, 1)
deplacer(3, 2)
deplacer(1, 2)
deplacer(1, 3)
deplacer(2, 3)
deplacer(2, 1)
deplacer(3, 1)
deplacer(2, 3)
deplacer(1, 2)
deplacer(1, 3)
deplacer(1, 3)
deplacer(2, 3)
```

```
["startSimu"],
["deplacer", "1", "2"],
["deplacer", "1", "3"],
["deplacer", "2", "3"],
["deplacer", "1", "2"],
["deplacer", "3", "1"],
["deplacer", "3", "2"],
["deplacer", "1", "2"],
["deplacer", "1", "3"],
["deplacer", "2", "3"],
["deplacer", "2", "1"],
["deplacer", "3", "1"],
["deplacer", "2", "3"],
["deplacer", "1", "2"],
["deplacer", "1", "3"],
["deplacer", "2", "3"],
["displayMsg", ""]
```

Recette secrète :

```
from robot import*
remplir(3)
transferer(3, 5)
remplir (3)
transferer(3,5)
vider(5)
transferer(3, 5)
remplir(3)
transferer(3, 5)
```

```
["startSimu"],
["remplir", "3"],
["transferer", "3", "5"],
["remplir", "3"],
["transferer", "3", "5"],
["vider", "5"],
["transferer", "3", "5"],
["remplir", "3"],
["transferer", "3", "5"],
["displayMsg", ""]
```

N1_C2. Répétitions d'instructions

1) Puniton :

```
for loop in range (135) :
    print("Je dois respecter le Grand Sorcier.")
```

```
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
Je dois respecter le Grand Sorcier.
```

Répétition : erreurs possibles :

Il est facile de se tromper dans les boucles lorsqu'on n'a pas l'habitude. Ainsi, si on oublie le « : » à la fin de la ligne, on obtient une erreur :

```
for loop in range(5)
    print("Bonjour !")
```

↳ `SyntaxError: invalid syntax`

Et si l'on oublie d'indenter, c'est-à-dire si on oublie les trois espaces, on obtient également une erreur :

```
for loop in range(5):
print("Bonjour !")
```

↳ `SyntaxError: expected an indented block`

Face à ce type d'erreur, on pensera donc à vérifier que le deux-points est bien présent et que l'indentation a été faite.

Indentation : la touche tabulation :

```
for loop in range(5):
    print("Bonjour !")
print("Au revoir !")
```

Le niveau d'indentation :

- **augmente (Tab) lorsque l'on rentre dans la boucle,**
- **et il diminue (Maj+Tab) lorsqu'on sort de la boucle.**

1) Réponds ! :

```
print(42)
```

Test 1 Succès 42

100 %

2) L'éclipse :

```
print(12581 - 11937)
```

Test 1 Succès Exécuté en 0,06 seconde.

100 %

3) Bonbons pour tout le monde :

```
print((25+30+27+22-8)*3)
```

Succès Exécuté en 0,06 seconde.

4) L'algoréathlon :

```
distance = 42
print(distance, end = " ")
print(2 * distance, end = " ")
print(3 * distance)
```

Succès 42 84 126

5) Cour de récréation :

```
distance = ((17 * 5) + (7 * 2) + (5 * 1) + (2 * 2))
print(distance * distance)
print(distance * 4)
```

Succès Exécuté en 0,06 seconde.

6) Une partie de cache-cache :

```
numero = 1
print(numero)
for loop in range(99):
    numero = numero +1
    print(numero)
print("J'arrive !")
```

Résultat :	91
	92
1	93
2	94
3	95
4	96
5	97
6	98
7	99
8	100
	J'arrive !

Variables : suppléments :

Plusieurs variables

```
largeur = 210
longueur = 297
surface = longueur * largeur
print(surface)
```

↳ 62370

Variable inexistante

Si on fait appel à une variable qui n'existe pas (ou pas encore), on obtient une erreur. Par exemple, le programme suivant définit une variable *longueur*, et tente ensuite d'afficher le contenu d'une variable nommée *largeur* qui n'a jamais été définie.

```
longueur = 297
print(largeur)
↳NameError: name 'largeur' is not defined
```

Il faut faire particulièrement attention au fait que les minuscules et majuscules ne sont pas considérées comme équivalentes.

Ainsi, la variable nommée *longueurFil* n'a strictement rien à voir avec la variable nommée *longueurfil*.

```
longueurFil = 10
print(longueurfil)
↳NameError: name 'longueurfil' is not defined
```

Si on rencontre une erreur de la forme `NameError: name 'xxxxx' is not defined`, on pensera donc à vérifier que l'on a bien initialisé la variable correspondante et que l'on a correctement saisi l'identifiant aux deux endroits.

Faute de frappe à l'affectation

Observez le programme suivant, dans lequel on a commis une faute de frappe sur la troisième ligne :

```
taille = 180
print(taille)
```

Au lieu d'afficher 180 puis 200 comme on voudrait, ce programme affiche deux fois de suite le nombre 180 :

↳ 180
180


```

ramasser()
for loop in range(3):
    gauche()
deposer()
for loop in range(4):
    droite()
ramasser()
for loop in range(4):
    gauche()
deposer()
for loop in range(5):
    droite()
ramasser()
for loop in range(5):
    gauche()
deposer()
for loop in range(6):
    droite()
ramasser()
for loop in range(6):
    gauche()
deposer()
for loop in range(7):
    droite()
ramasser()
for loop in range(7):
    gauche()
deposer()
for loop in range(8):
    droite()
ramasser()
for loop in range(8):
    gauche()
deposer()
for loop in range(9):
    droite()
ramasser()
for loop in range(9):
    gauche()
deposer()
for loop in range(10):
    droite()
ramasser()
for loop in range(10):
    gauche()
deposer()

```

12) Construction d'une pyramide :

```

cube1 = 1
ligne3 = cube1 + 2
cube2 = ligne3 * ligne3 * ligne3
ligne5 = ligne3 + 2
cube3 = ligne5 * ligne5 * ligne5
ligne7 = ligne5 + 2
cube4 = ligne7 * ligne7 * ligne7
ligne9 = ligne7 + 2
cube5 = ligne9 * ligne9 * ligne9
ligne11 = ligne9 + 2
cube6 = ligne11 * ligne11 * ligne11
ligne13 = ligne11 + 2
cube7 = ligne13 * ligne13 * ligne13
ligne15 = ligne13 + 2
cube8 = ligne15 * ligne15 * ligne15
ligne17 = ligne15 + 2
cube9 = ligne17 * ligne17 * ligne17

```

Test 1 Succès Exécuté en 0,06 seconde.

100 %

```
cubetotal = cube1 + cube2 + cube3 + cube4 + cube5 + cube6 + cube7 + cube8 + cube9
print(cubetotal)
```

13) Table de multiplication :

```
colonne = 1
ligne = 2
for loop in range(20):
    print(1 * colonne, end = " ")
    print(2 * colonne, end = " ")
    print(3 * colonne, end = " ")
    print(4 * colonne, end = " ")
    print(5 * colonne, end = " ")
    print(6 * colonne, end = " ")
    print(7 * colonne, end = " ")
    print(8 * colonne, end = " ")
    print(9 * colonne, end = " ")
    print(10 * colonne, end = " ")
    print(11 * colonne, end = " ")
    print(12 * colonne, end = " ")
    print(13 * colonne, end = " ")
    print(14 * colonne, end = " ")
    print(15 * colonne, end = " ")
    print(16 * colonne, end = " ")
    print(17 * colonne, end = " ")
    print(18 * colonne, end = " ")
    print(19 * colonne, end = " ")
    print(20 * colonne)
    colonne = colonne + 1
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120
7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140
8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136 144 152 160
9 18 27 36 45 54 63 72 81 90 99 108 117 126 135 144 153 162 171 180
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209 220
12 24 36 48 60 72 84 96 108 120 132 144 156 168 180 192 204 216 228 240
13 26 39 52 65 78 91 104 117 130 143 156 169 182 195 208 221 234 247 260
14 28 42 56 70 84 98 112 126 140 154 168 182 196 210 224 238 252 266 280
15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240 255 270 285 300
16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320
17 34 51 68 85 102 119 136 153 170 187 204 221 238 255 272 289 306 323 340
18 36 54 72 90 108 126 144 162 180 198 216 234 252 270 288 306 324 342 360
19 38 57 76 95 114 133 152 171 190 209 228 247 266 285 304 323 342 361 380
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400
```

Suivre l'évolution des variables d'un programme :

```
nbPersonnes = 3
nbJours = 4
prixSéjour = 0
prixJournée = 1
for loop in range(nbJours):
    prixSéjour = prixSéjour + prixJournée
    prixJournée = prixJournée + 2
prixSéjour = prixSéjour + 14
print("Prix par personne :", prixSéjour // nbPersonnes)
```

Le résultat est donc :

↳ Prix par personne : 10

N1_C4. Lecture de l'entrée

Des programmes interactifs :

```
nom <- lireLigne()
année <- lireEntier()
Afficher nom, ", vous aviez ", 2000 - année, " ans en l'an 2000 !"
```

Entrée :

```
Loïc
1986
```

1) Récoltes :

```
longueur = int(input())
print(longueur * longueur * 23)
```

Test 10 Succès Exécuté en 0,06 seconde.

2) Retraite spirituelle :

```
Jour = int(input())
print(Jour * 3600 * 16)
```

Test 11 Succès Exécuté en 0,06 seconde.

3) Âge des petits-enfants :

Test 3 Succès Exécuté en 0,06 seconde.

```

âgeCadet = int(input())
âgeAîné = int(input())
différence = âgeAîné - âgeCadet
print(différence)

```

Représentation de l'entrée d'un programme :

```

nombre = int(input())
print(nombre - 5)
print(nombre * 2)

```

↳ 9
↳ 4
18

Erreur si l'on ne donne pas un entier :

Imaginons que le programme demande un entier mais que l'utilisateur fournisse un texte, par exemple « coucou ». Le programme déclare alors une erreur, car « coucou » ne peut pas être interprété comme un nombre.

```

taille = int(input())
print(taille)

```

↳ coucou
↳ ValueError: invalid literal for int() with base 10: 'coucou'

4) Encore des punitions :

```

nombre = int(input())
for loop in range (nombre):
print("Je dois suivre en cours")

```

Test 1 Succès Exécuté en 0,06 seconde. 100 %

5) Graduation de thermomètres :

```

nombre = int(input())
nombre2 = int(input())
for loop in range (nombre2 - nombre +1):
print(nombre)
nombre = nombre + 1

```

TOTAL Succès Félicitations, vous avez résolu ce problème ! 100 %

6) Jeu de calcul mental :

```

nbNombres = int(input())
tour = 0
départ = 66
for loop in range(nbNombres):
tour = tour + 1
départ = tour * départ
print (départ)

```

Exemple 1	Exemple 2
entrée :	entrée :
4	1
sortie :	sortie :
66	66
132	
396	
1584	

7) La Grande Braderie :

```

positionDepart = int(input())
largeurEmplacement= int(input())
nbVendeurs = int(input())
print(positionDepart)
for loop in range(nbVendeurs):
positionDepart = positionDepart + largeurEmplacement
print(positionDepart)

```

Exemple 1

entrée :

10
5
3

sortie :

10
15
20
25

Exemple 2

entrée :

12
8
5

sortie :

12
20
28
36
44
52

Lecture d'entiers : autre erreur possible :

↳ 11
22
↳ 1122

```

valeur1 = input()
valeur2 = input()
print(valeur1 + valeur2)

```

8) Bétail :

```

A = int(input())
B = int(input())
C = int(input())
D = int(input())
E = int(input())
F = int(input())
G = int(input())
H = int(input())
I = int(input())
J = int(input())
K = int(input())
L = int(input())
M = int(input())
N = int(input())
O = int(input())
P = int(input())
Q = int(input())
R = int(input())
S = int(input())
T = int(input())
print(A + B + C + D + E + F + G + H + I + J + K + L + M + N + O + P + Q + R + S + T)

```

Exemple 1

entrée :

1
2
3
4
5
6
7
8
9
10
10
9
8
7
6
5
4
3
2
1

sortie :

110

Exemple 2

entrée :

1
1
1
1
1
1
1
1
1
1
2
2
2
2
2
2
2
2
2
2
2

sortie :

30

9) Socles pour statues :

```

niveauMax = int(input())
niveauMin = int(input())
socle = 0
for loop in range(niveauMax - niveauMin + 1):
    socle = socle + niveauMax * niveauMax
    niveauMax = (niveauMax - 1)
print(socle)

```

entrée :

7
3

sortie :

135

Instructions condensées :

```

longueur = int(input())
largeur = int(input())
périmètre = (longueur + largeur) * 2
aire = longueur * largeur
print(périmètre, end = " ")
print(aire)

```

↳ 12
7
↳ 38 84

```

longueur = int(input())
largeur = int(input())
périmètre = (longueur + largeur) * 2
aire = longueur * largeur
print(périmètre, aire)

```

```

âge = 20
tempsProgrammation = 8
print("J'ai", âge, "ans et je programme depuis", tempsProgrammation, "ans")

```

10) Le plus beau Karva :

```

nbCompétition = int(input())
for iValeur in range(nbCompétition):
    poids = int(input())
    âge = int(input())
    cornes = int(input())
    garrot = int(input())
    calcul = (cornes * garrot + poids)
    print(calcul)

```

entrée :

2
100
5
25
90
300
10
15
120

sortie :

2350
2100

Erreur si on lit trop de choses :

```
largeur = int(input())
hauteur = int(input())
profondeur = int(input())
print(largeur * hauteur * profondeur)
```

```
↳ 4
   5
```

```
↳ Traceback (most recent call last):
  File "./run/exe", line 3, in
    profondeur = int(input())
EOFError: EOF when reading a line
```

Portée d'une variable :

```
nbValeurs = int(input())
for iValeur in range(nbValeurs):
    laValeur = int(input())
print(laValeur)
```

Ainsi, la variable *laValeur*, qui reçoit des valeurs au sein de la boucle, peut être réutilisée après la boucle. Cela peut s'avérer dangereux : en effet, que se passe-t-il dans le programme si 0 est saisi pour *nbValeurs* ? Dans ce cas, *laValeur* n'est jamais initialisée, et on obtient l'erreur suivante :

```
↳ Traceback (most recent call last):
  File "./run/exe", line 4, in <module>
    print(laValeur)
NameError: name 'laValeur' is not defined
```

N1_C5. Tests et conditions

1) Transport des bagages :

```
nbPaquets = int(input())
poidsPaquet = int(input())
if nbPaquets * poidsPaquet > 105:
    print("Surcharge !")
```

2) Bornes kilométriques :

```
numéroMatin = int(input())
numéroSoir = int(input())
écart = numéroSoir - numéroMatin
if écart < 0:
    écart = -écart
print(écart)
```

3) Tarifs dégressifs :

```
heure = int(input())
prix = 10 + 5 * heure
if prix > 53:
    prix = 53
print(prix)
```

4) Bagarre générale :

```
superficieArignon = int(input())
superficieEvaran = int(input())
if superficieArignon > superficieEvaran + 10:
    print("La famille Arignon a un champ trop grand")
if superficieEvaran > superficieArignon + 10:
    print("La famille Evaran a un champ trop grand")
```

5) Tarif du bateau :

```
âge = int(input())
if âge < 21:
    print("Tarif réduit")
else:
    print("Tarif plein")
```

6) Traversée du pont :

```
premierDé = int(input())
secondDé = int(input())
somme = premierDé + secondDé
if somme >= 10:
    print("Taxe spéciale !")
    print(36)
else:
    print("Taxe régulière")
    print(2 * somme)
```

7) Concours de tir à la corde :

```
nbPersonnes = int(input())
totalÉquipe1 = 0
totalÉquipe2 = 0
for loop in range(nbPersonnes):
    poids1 = int(input())
    poids2 = int(input())
    totalÉquipe1 = totalÉquipe1 + poids1
    totalÉquipe2 = totalÉquipe2 + poids2
if totalÉquipe1 > totalÉquipe2:
    print("L'équipe 1 a un avantage")
else:
    print("L'équipe 2 a un avantage")
print("Poids total pour l'équipe 1 :", totalÉquipe1)
print("Poids total pour l'équipe 2 :", totalÉquipe2)
```

8) Mot de passe du village :

```
tentative = int(input())
if tentative == 64741:
    print("Bon festin !")
else:
    print("Allez-vous-en !")
```

N1_C6. Structures avancées

Structures imbriquées :

```
cible = int(input())
if cible >= 0:
    for numéro in range(cible + 1):
        print(numéro)
else:
    print("Rien à faire")
```

1) Villes et villages :

```
cible = int(input())
tour = 0
for loop in range(cible):
    ville = int(input())
    if ville > 10000:
        tour = tour + 1
print(tour)
```

2) Planning de la journée :

Solution de Microsoft COPILOT :	Solution dans France ioi :
depart = int(input("Enter your position: ")) nbVillage = int(input("Enter number of villages: ")) comptage = 0	posActuelle = int(input()) nbVillages = int(input()) nbAccessibles = 0

```
for _ in range(nbVillage):
    village_position = int(input("Enter village position: "))
    if abs(village_position - depart) <= 50:
        comptage += 1
print("Number of villages within 50 units:", comptage)
```

```
for loop in range(nbVillages):
    posVillage = int(input())
    ecart = posActuelle - posVillage
    if ecart < 0:
        ecart = -ecart
    if ecart <= 50:
        nbAccessibles = nbAccessibles + 1
print(nbAccessibles)
```

3) Étape la plus longue :

4) Calcul des dénivelées :

5) Type d'arbres :

6) Tarifs de l'auberge :

7) Protection du village :

8) Le juste prix :

N1_C7. Conditions avancées, opérateurs booléens

1) Espion étranger :

2) Maison de l'espion :

3) Nombre de jours dans le mois :

4) Amitié entre gardes :

5) Nombre de personnes à la fête :

6) Casernes de pompiers :

7) Personne disparue :

8) La grande fête :

9) L'espion est démasqué :

10) Zones de couleurs :

N1_C8. Répétitions conditionnées

1) Département de médecine : contrôle d'une épidémie :

2) Administration : comptes annuels :

3) Département de pédagogie : le « c'est plus, c'est moins » :

4) Département d'architecture : construction d'une pyramide :

5) Département de chimie : mélange explosif :

NIVEAU 2

N2_C1. Nombres à virgules et autres outils

1) Origami :

2) Conversions de distances :

3) Comparatif de prix :

4) Moyenne des notes :

A – Faire des arrondis (inférieur et supérieur)

1) Augmentation de la population :

2) Construction de maisons :

B – Faire des arrondis (au plus proche)

1) Soirée orageuse :

2) Augmentation des taxes :

C – Arithmétique de base

1) Achat de livres :

2) Une belle récolte :

3) La roue de la fortune :

N2_C2. Découverte des tableaux

1) Préparation de l'onguent :

2) Liste de courses :

3) Grand inventaire :

4) Étude de marché :

5) Répartition du poids :

6) Visite de la mine :

7) Journée des cadeaux :

8) Course à trois jambes :

9) Banquet municipal :

10) Choix des emplacements :

A – Chaînes complètes

1) Petites fiches et gros travail :

2) Priorité alphabétique :

3) Une ligne sur deux :

4) Résumés de livres :

5) Lire ou ne pas lire, telle est la question :

B – Mots

1) Fiches d'inscription :

2) Analyse de fréquence :

C – Caractères

1) Impression d'étiquettes :

2) Écriture en miroir :

3) Inscription d'étudiants :

4) ngms sns vlls :

5) La bataille :

6) Analyse d'une langue :

7) Sans espaces :

1) Code secret deux fois :

2) Deux codes secrets :

3) Dentelle :

4) Motif rectangulaire :

5) Le plus petit de deux entiers :

6) Phénomène numérique :

7) Distance euclidienne :

8) Formes creuses :

9) Convertisseur d'unités :

N2_C5. Programmer sur son ordinateur

NIVEAU 3

N3_C1.

N3_C2.

N3_C3.

N3_C4.

N3_C5.

N3_C6.

N3_C7.

N3_C8.

N3_C9.

N3_C10.

N3_C11.

NIVEAU 4

N4_C1.

N4_C2.

N4_C3.

N4_C4.

N4_C5.

N4_C6.

N4_C7.

N4_C8.

N4_C9.

NIVEAU 5

N5_C1.

N5_C2.

N5_C3.

N5_C4.

N5_C5.

N5_C6.

N5_C7.

N5_C8.

N5_C9.

NIVEAU 6

N6_C1.

N6_C2.

N6_C3.

N6_C4.

N6_C5.

N6_C6.

N6_C7.