

```

1 #####
2 ### Le 02/11/2021 ###
3 ### Correction DS01: Introduction à la programmation ###
4 ### 1 SPE NSI4 ###
5 #####
6
7
8 #####
9 # Exercice 1: (10 points) #
10 #####
11
12 # 1.1. Programmer une instruction qui affiche un texte entré
13 # par l'utilisateur depuis le clavier.
14 # -----
15
16 print(input('rentrez un texte, il sera affiché'))
17
18
19 # 1.2. Ecrire ce que va afficher le programme:
20 # -----
21
22 a = 3 # a = 3
23 c = 2 # c = 2
24 a = a + 1 # a = 4
25 b = a # b = 4
26 b = b * 2 # b = 8
27 c = a ** b # c = 4 ** 8 = 65536
28 print(a) # 4
29 print(b) # 8
30 print(c) # 65536 (4 puissance 8)
31
32 '''
33 Le programme va afficher:
34 4
35 8
36 65536 (4 puissance 8)
37 '''
38
39
40 # 1.3. Ecrire ce que va afficher le programme:
41 # -----
42
43 a = 3
44 c = 2
45 a = a + 1
46 b = a
47 if a < c:
48     a = b + 1
49 elif a == b:
50     a = b
51 else:
52     a = b - 1
53 b = b * 2
54 c = a ** b
55 print(a)
56 print(b)
57 print(c)
58
59 '''
60 Le programme va afficher:
61 4
62 8
63 65536 (4 puissance 8)
64 '''
65
66
67 # 1.4. Ecrire un programme qui affiche 100 fois votre nom,
68 # chaque fois sur une ligne différente.
69 # -----
70
71 for i in range(100):
72     print('Mathex')
73
74
75 # 1.5. Ecrire un programme qui:
76 # 1. Demande à l'utilisateur un nombre entre 0 et 20
77 # 2. Si ce nombre est inférieur à 15, lui dit 'Pas assez' et revient en 1.
78 # 3. Si ce nombre est supérieur ou égal à 15, lui dit 'Bravo' et se termine
79 # -----
80
81 n = int(input('rentrez un nombre entre 0 et 20: '))
82 while n < 15:
83     print('Pas assez')
84     n = int(input('rentrez un nombre entre 0 et 20: '))
85 print('Bravo')
86
87 # ou encore:
88 n = 0
89 while n < 15:
90     print('Pas assez')
91     n = int(input('rentrez un nombre entre 0 et 20: '))
92 print('Bravo')
93
94
95 # 1.6. Programmer une fonction qui prend deux nombres en paramètres
96 # et renvoie leur somme
97 """

```

```

97 # -----
98
99 def somme(a, b):
100     return a + b
101
102
103 # 1.7. Ecrire ce que va afficher la programme ci-dessous:
104 #     a = [ 1 , 2 , 3 , 4 ]
105 #     b = [ a , a , 1 , 2 ]
106 #     print( a[1] )
107 #     print( b[1][3] )
108 # -----
109
110 a = [ 1 , 2 , 3 , 4 ]
111 b = [ a , a , 1 , 2 ]
112 print( a[1] ) # 2 (attention le premier indice est 0)
113 print( b[1][3] ) # 4 ( a[3] )
114
115 '''
116 Le programme va afficher:
117 2
118 4
119 '''
120
121
122 #####
123 #                               Exercice 2: ( 5 points)                               #
124 #####
125
126 # 2.1. Ecrire un programme qui affiche la table de multiplication de 3:
127 # -----
128
129 for i in range(1, 10):
130     print(3,'x',i,'=',3*i)
131
132
133 # 2.2. Ecrire un programme qui affiche en lignes les tables
134 # de multiplication de 1 à 9:
135 # -----
136
137 for j in range(1,10):
138     for i in range(1, 10):
139         print(j,'x',i,'=',j*i)
140
141 # 2.3. Ecrire un programme qui affiche en tableau les tables
142 # de multiplication de 1 à 9:
143 # -----
144
145 for i in range(1,10):
146     ligne = ''
147     for j in range(1, 10):
148         ligne = ligne + str(j) + 'x' + str(i) + '=' + str(j*i) + ' '
149     print(ligne) # manque l'alignement des colonnes
150
151 # on pourrait utiliser les paramètres et les options d'affichage
152 # ( end=' ', % nombre caractères, ...)
153
154
155 # 2.4. Ecrire une fonction qui prend en paramètres
156 # un entier n et retourne une liste
157 # dont chaque élément est le tete d'une ligne de la table de n
158 # ['3x1=3', '3x2=6', '3x3=9', '3x4=12', '3x5=15', '3x6=18', '3x7=21', '3x8=24',
159 # '3x9=27']
159 # -----
160
161 def tableMultiplication(n):
162     table = []
163     for i in range (1, 10):
164         table.append(str(n) + 'x' + str(i) + '=' + str(n*i))
165     return table
166
167 print(tableMultiplication(3))
168
169
170 # En option (++) : Utiliser cette fonction pour afficher
171 # en tableau les tables de multiplication
172 # -----
173
174 def tablesMultiplication(a, b):
175     table = []
176     for i in range (a, b+1):
177         table.append(tableMultiplication(i))
178     return table
179
180 def afficheTable(table):
181     max_car = 5 # ou le déterminer
182     for j in range(len(table[0])):
183         ligne = ''
184         for i in range(len(table)):
185             ligne += table[i][j] + ' '
186             if len(table[i][j]) == max_car:
187                 ligne += ' '
188         print(ligne)
189
190 afficheTable(tablesMultiplication(1,9))
191
192
193 #####

```

```

194 # Exercice 3: ( 5 points) #
195 #####
196
197
198
199 def manche( coup_1, coup_2):
200     '''
201     compare les coups des deux joueurs ('R', 'P' ou 'C')
202     et renvoie:
203         0 si égalité
204         1 si coup_1 gagne coup_2
205        -1 si coup_2 gagne coup_1
206     rappel des règles: R bat C, P bat R et C bat P
207     '''
208     # égalité si même coup
209     if coup_1 == coup_2:
210         return 0
211
212     # Pierre(R) gagne contre Ciseaux
213     elif coup_1 == 'R':
214         if coup_2 == 'C':
215             return 1
216         else :
217             return -1
218
219     # Papier gagne contre Pierre(R)
220     elif coup_1 == 'P':
221         if coup_2 == 'R':
222             return 1
223         else :
224             return -1
225
226     # Ciseaux gagne contre Papier
227     elif coup_1 == 'C':
228         if coup_2 == 'P':
229             return 1
230         else :
231             return -1
232
233 def coupJoueur(nom_joueur):
234     '''
235     demande à un joueur son coup (parmi 'R', 'P' ou 'C')
236     lui redemande s'il est incorrect
237     et le renvoie s'il est correct
238     '''
239     correct = False
240     while correct == False:
241         resultat = input(nom_joueur + ': Pierre (R), Papier (P) ou Ciseaux (C) ?
242     ')
243         if resultat in ['R', 'P', 'C']:
244             correct = True
245         return resultat
246
247 def match(n):
248     '''
249     demande pour chaque manche à chaque joueur son coup (parmi 'R', 'P' ou 'C')
250     et renvoie :
251         True si le joueur 1 gagne le match (sur n points)
252         False si le joueur 2 gagne le match (sur n points)
253     '''
254     point_1 = 0
255     point_2 = 0
256     while point_1 < n and point_2 < n:
257         coup_1 = coupJoueur('Joueur 1')
258         coup_2 = coupJoueur('Joueur 2')
259         resultat = manche( coup_1, coup_2)
260         if resultat == 1:
261             point_1 += 1
262
263         elif resultat == -1:
264             point_2 += 1
265         print('joueur 1:', point_1, 'joueur 2:', point_2)
266     if point_1 >= n:
267         return True
268     else:
269         return False
270
271 win_1 = match(3)
272 if win_1:
273     print('Le joueur 1 a gagné')
274 else:
275     print('Le joueur 2 a gagné')
276
277
278 #####
279 # Fin de la correction #
280 #####

```