Approfondissement

débranché

papier & crayon

1. Les nombres et opérations

1 Compléter le tableau en donnant les résultats correspondant aux opérations écrites en langage Python.

Opération saisie	5*7	18/4	3**2	8//5	7%3	sqrt(3) (arrondir au centième)	2*7+1	2*(7+1)	sqrt(11)**2
Résultat									

2 On considère le script suivant.

```
x = 1
for i in range(1,5) :
    x = 0.5*(x + 2/x)
```

1. Compléter le tableau d'état pour qu'il donne toutes les étapes de la boucle. Arrondir les résultats au millième.

i		1			
x	1				

2. Quelle(s) modification(s) faut-il apporter au script pour qu'il affiche les résultats arrondis au millième ?

.....

3 Roxane veut s'acheter un smartphone qui coûte 139,99 €. Elle a 10 € dans sa tirelire. Chaque mois, elle dépense un cinquième de sa tirelire et, à la fin du mois, gagne 40 € d'argent de poche en faisant du babysitting.

Roxane a commencé à écrire l'algorithme ci-contre en langage naturel.

1. *tirelire* ← 10
2. *mois* ← 0
3. Tant que *tirelire* < .....
4. *tirelire* ← .....
5. *mois* ← .....
6. Fin Tant que

1. Expliquer le rôle des deux variables *tirelire* et *mois*.

.....

.....

.....

.....

.....

2. Compléter l'algorithme en langage naturel de Roxane puis l'écrire ci-contre en langage Python.

3. Compléter le tableau d'état en utilisant la calculatrice pour faire les calculs intermédiaires. Arrondir les résultats au centime d'euro. Le symbole V signifie que la condition de la boucle est vraie.

Tirelire	10						
Mois	0						
Condition	V						

4. Quelle est la valeur de la variable *mois* après exécution du script ? Que signifie-t-elle ?

.....

.....

4 **1.** Taper dans la console Python les instructions ci-contre et noter les valeurs affichées.

2. Comment peut-on expliquer ces résultats ?

```
>>> from math import sqrt
```

a. `>>> sqrt(3)**2` .....

b. `>>> sqrt(22)**2` .....

.....

.....

5 On appelle « nombre parfait » un nombre qui est égal à la somme de tous ses diviseurs sauf lui-même. Par exemple, 6 est un nombre parfait car ses diviseurs sont 1, 2, 3 et 6, et 6 = 1 + 2 + 3.
On considère la fonction ci-contre.

```
1 def somme_div(n):
2     S=0
3     for k in range (1,n):
4         if n%k==0:
5             S=S+k
6     return S
```

1. Expliquer la ligne 4.

.....

.....

.....

.....

2. Qu'obtient-on lorsqu'on écrit dans la console somme_div(12) ? Quelle est la signification de ce résultat ?

.....

.....

3. Recopier et compléter la fonction ci-contre dans l'éditeur Python afin qu'elle utilise la fonction précédente somme_div et retourne « True » si n est parfait ou « False » sinon.

```
1 def parfait(n):
2     S=.....
3     if S==...:
4         resultat=.....
5     else:
6         resultat=.....
7     return resultat
```

4. Quel est le résultat affiché lorsqu'on saisit dans la console Python :
parfait(16) ? ..... parfait(496) ? .....

6 On considère la fonction affine f définie par $f(x) = 2{,}3x - 4$.
Valentin et Yanis veulent écrire une fonction qui détermine si un point $A(x_A\,;\,y_A)$ appartient ou non à la représentation graphique $\mathscr{C}$ de la fonction f.

Valentin propose le script suivant.

```
1 def appartient(x_A,y_A) :
2     if y_A == 2.3*x_A-4 :
3         return("A appartient à C")
4     else :
5         return("A n'appartient pas à C")
```

1. Yanis lui dit que son script ne fonctionne pas. Il lui suggère de l'essayer avec le point $A(1\,;\,-1{,}7)$ qui appartient à $\mathscr{C}$. A-t-il raison ? Pourquoi ?

.....

.....

.....

.....

2. Yanis propose alors la fonction suivante.

```
1 from numpy import isclose
2 def appartient(x_A,y_A) :
3     if isclose(2.3*x_A-4,y_A) :
4         return("A appartient à C")
5     else :
6         return("A n'appartient pas à C")
```

a. Copier ou ouvrir la fonction appartient dans l'éditeur Python et la tester avec le point $A(1\,;\,-1{,}7)$. Fonctionne-t-elle ?

.....

b. Expliquer l'instruction isclose de la ligne 3.

.....

.....

.....

.....