

Exercice 1 :

1) Expliquer le principe de l'algorithme ci-contre. Que permet-il de faire ?

```
Saisir a
b ← a/13
c ← le quotient de la division
      euclidienne de a par 13
Si b = c
  Alors afficher "True"
Sinon
  Afficher "False"
Fin Si
```

2) Ce même algorithme peut se traduire par le programme ci-dessous.

Quelles valeurs obtient-on pour b et c lorsqu'on saisit a = 182 au départ ? Qu'affiche l'algorithme en sortie dans ce cas.

```
def div(a):
    b=a/13
    c=a//13
    if b==c:
        return True
    else:
        return False
Commentaires :
"==" est le symbole d'égalité ; "=" celui d'affectation.
```

3) a) Modifier le programme dans le but de vérifier si un nombre est divisible par 29.

b) Les nombres suivants sont-ils divisibles par 29 ?

565 – 6785 – 646 195 034 – 1 970 659 794

Exercice 2 :

Ecrire un programme permettant de vérifier si un nombre donné est divisible par 13 en effectuant un test sur le reste de la division de ce nombre par 13.

Correction :

1)

Ce algorithme sert à envoyer deux valeurs booléennes (vrai, faux) vrai si le nombre saisi est un diviseur de 13 et faux sinon

a	b	c
28	2,15	2

2,15 est différent de 2 donc le nombre ne divise pas 13

2) La commande « def » sert à déclarer une fonction qui permet de ne pas réécrire tout le programme à chaque test.

```
def div(a):
    b=a/13
    c=a//13
    if b==c:
        return True
    else:
        return False
print(div(182))
print(div(18))
print(div(114))
```

3)a)b)

```
def div(a):
    b=a/29
    c=a//29
    if b==c:
        return True
    else:
        return False
print(div(565))
print(div(6785))
print(div(646195034))
print(div(1970659794))
```

```
*** Console de processus distant Réinitialisée ***
False
False
False
True
>>>
```

Exercice 2 :

Ecrire un programme permettant de vérifier si un nombre donné est divisible par 13 en effectuant un test sur le reste de la division de ce nombre par 13.

Correction :

on utilise l'opérateur modulo % qui renvoie le reste d'une division euclidienne

```
def div(a):
    if a%13==0:
        return True
    else:
        return False
```

```
Saisir x
Saisir y
Si x < 5y
    Alors x ← 10x
Sinon
    y ← 10y
Fin Si
Afficher xy
```

Exercice 3 :

Dans le programme ci-dessous traduisant l'algorithme ci-contre, les instructions conditionnelles ont été supprimées.

```
def fonction(x,y):
    x<5*y:
    x=10*x
    y=10*y
    return x*y
```

- 1) Corriger en complétant le programme par les instructions conditionnelles manquantes.
- 2) Tester ce programme pour x = 5 et y = 9. Même question pour x = 12 et y = 2.

Correction :

1)

```
def fonction(x,y):
    if x<5*y:
        x=10*x
    else:
        y= 10*y
    return x*y
x= float(input('Entrer x: '))
y= float(input('Entrer y: |'))
print(fonction(x,y))
```

2)

```
*** Console de processus distant Réinitialisée ***
Entrer x: 5
Entrer y: 9
450.0
>>>
*** Console de processus distant Réinitialisée ***
Entrer x: 12
Entrer y: 2
240.0
>>>
```

Exercice 5 :

Dans le programme ci-contre, les affichages en sortie de l'algorithme ont été supprimés.

1) Quel problème permet de résoudre cet algorithme ?

2) Compléter le programme par les affichages en sortie manquants.

3) Tester ce programme pour différentes valeurs de a et b.

```
def signe(a,b):
    if a>0:
        if b>0:
            print("le produit axb est ...")
        else:
            print("le produit axb est ...")
    else:
        if b>0:
            print("le produit axb est ...")
        else:
            print("le produit axb est ...")
```

Correction :

1) le programme a comme d'afficher le signe d'une produit de deux nombres réels.

```
def signe(a,b):
    if a>0:
        if b>0:
            print("le produit axb est strictement positif")
        else:
            print("le produit axb est strictement négatif")
    else:
        if b>0:
            print("le produit axb est strictement négatif")
        else:
            print("le produit axb est strictement positif")
    signe(2,3)
    signe(2,-3)
    signe(-2,3)
    signe(-2,-3)
```

3)

```
*** Console de processus distant Réinitialisée ***
le produit axb est strictement positif
le produit axb est strictement négatif
le produit axb est strictement négatif
le produit axb est strictement positif
>>>
```

Exercice 6 :

Écrire et tester un programme qui demande en entrée à un client le montant total de ses achats.

En fonction de la somme dépensée, le programme affiche en sortie le prix à payer :

- Si la somme dépensée est strictement inférieure à 75 €, il obtient 5 % de remise.

– Si la somme dépensée est supérieure à 75 €, il obtient 8 % de remise.

Correction :

```
montant= float(input('Entrer la montant total de vos achats'))
if montant>75:
    print('Le prix à payer est: ', montant*0.95)
else:
    print('Le prix à payer est: ', montant*0.92)
```