

Exercices d'algorithmie sur l'arithmétique

Exercice 1 : Afficher les **diviseurs d'un nombre** et le **nombre de ses diviseurs**

Construire un algorithme puis un programme en python permettant d'afficher tous les diviseurs d'un nombre. Vous devrez utiliser au moins une fonction dans votre programmation.

On rappelle les instructions suivantes donnant le quotient et le reste de la division euclidienne :

$$\text{Quotient : } 17 // 3 = 5$$

$$\text{Reste : } 17 \% 3 = 2$$

Exercice 2 : **Identifier un nombre premier**

Construire un algorithme puis un programme en python permettant de dire si le nombre saisi est premier. Vous devrez utiliser au moins une fonction dans votre programmation.

Exercice 3 : **Calcul du pgcd de deux nombres**

Construire un algorithme puis un programme en python permettant d'afficher le pgcd de deux nombres. Vous devrez utiliser au moins une fonction dans votre programmation.

Exercice 4 : Afficher une **liste des premiers nombres premiers**

Saisir un nombre puis afficher tous les nombres premiers inférieurs à ce nombre.

Exercice 5 : Afficher la **liste des carrés des nombres impairs**

Construire un algorithme puis un programme en python permettant d'afficher, pour un nombre donné, la liste des carrés des nombres impairs inférieurs à ce nombre.

Vous devrez utiliser au moins une fonction dans votre programmation.

CORRIGE – Notre Dame de La Merci – Montpellier

Exercice 1 : Afficher les diviseurs d'un nombre et le nombre de ses diviseurs
Construire un algorithme puis un programme en python permettant d'afficher tous les diviseurs d'un nombre. Vous devrez utiliser au moins une fonction dans votre programmation.

A partir d'un nombre entier n saisi, on va regarder le reste de la division euclidienne de ce nombre par tous les entiers de 1 à n .
 →chaque fois que le reste est égal à zéro, on affiche le diviseur testé et on incrémente de 1 la variable qui compte le nombre de diviseurs.

Algorithme :

```
Saisir un nombre  $n$ 
Initialiser une variable nb_diviseurs à 0
Pour  $i$  variant de 1 à  $n$ 
    Si le reste de la division euclidienne de  $n$  par  $i$  est égal à 0
        Afficher  $i$ 
        Augmenter la variable nb_diviseurs de 1
```

Programme python :

```
def diviseurs_nombre(n):
    nb_diviseurs=0
    for i in range(1,n+1):
        if n%i==0:
            print(i)
            nb_diviseurs+=1
    return(nb_diviseurs)

n=int(input("Veuillez saisir un nombre : "))
nbd=diviseurs_nombre(n)
print("Le nombre",n, "possède", nbd, "diviseurs")
```

Exemple : on saisit 10

```
1
2
5
10
→Le nombre 10 possède 4 diviseurs
```

Exercice 2 : Identifier un nombre premier
Construire un algorithme puis un programme en python permettant de dire si le nombre saisi est premier. Vous devrez utiliser au moins une fonction dans votre programmation.

A partir d'un nombre entier n saisi, on va regarder le reste de la division euclidienne de ce nombre par tous les entiers de 2 à $n-1$.
 →dès que le reste est différent de zéro, on sort de la boucle d'étude car le nombre n n'est pas premier,
 →si l'on arrive en fin de boucle et que tous les restes ont été égaux à zéro, le nombre est premier
 On devra utiliser une boucle tant_ que car on n'est pas obligé de tester toutes les divisions

Algorithme :

Saisir un nombre n
 Initialiser une variable i à 2
 Initialiser une variable nbp à 0
 Tant que $i < n$
 Si le reste de la division euclidienne de n par i est égal à 0
 Affecter à nbp la valeur 1
 Augmenter le compteur i de 1
 Au final, si la variable nbp n'a pas été modifiée et est toujours égale à 0
 Afficher « le nombre est premier »
 Sinon
 Afficher « le nombre n'est pas premier »

Programme python :

```
def nb_premier(n):
    i=2
    test_premier=0
    while i < n:
        if n%i == 0:
            test_premier=1
        i+=1
    return(test_premier)

n=int(input("Veuillez saisir un nombre : "))
nbp=nb_premier(n)
if nbp==0:
    print("Le nombre",n, "est premier")
else:
    print("Le nombre",n, "n'est pas premier")
```

Exemple : on saisit 10
 → Le nombre 10 n'est pas premier

Exemple : on saisit 7
 → Le nombre 7 est premier

Exercice 3 : Calcul du pgcd de deux nombres

Construire un algorithme puis un programme en python permettant d'afficher le pgcd de deux nombres.
 Vous devrez utiliser au moins une fonction dans votre programmation.

Un exemple permet de comprendre la démarche à suivre : cherchons le pgcd de 160 et 96

$$160 = 96 \times 1 + 64$$

$$96 = 64 \times 1 + 32$$

$$64 = 32 \times 2 + 0 \quad \rightarrow \text{le pgcd est } 32.$$

Algorithme :

Saisir deux nombres a et b
 Si $a < b$, inverser les valeurs de a et de b pour simplifier la programmation.
 Tant que le reste de la division euclidienne de a par b est différent de zéro
 Attribuer à a la valeur de b
 Attribuer à b la valeur du reste de la division euclidienne de a par b
 Afficher b

Programme python :

```
def calcul_pgcd(a,b):
    if b>a:
        a,b=b,a
    while a%b != 0:
        c=a%b
        a=b
        b=c
    return(b)

a=int(input("Veuillez saisir le premier nombre : "))
b=int(input("Veuillez saisir le deuxième nombre : "))
if a==0 or b==0:
    print("Un des nombres est nul, veuillez relancer le programme")
else :
    pgcd=calcul_pgcd(a,b)
    print("Le pgcd de ces nombres est",pgcd)
```

Exemple : on saisit 160 et 96
 → Le pgcd de ces nombres est 32
 on saisit 63 et 0
 →Un des nombres est nul, veuillez relancer le programme

Exemple : on saisit 96 et 160
 → Le pgcd de ces nombres est 32

Exercice 4 : Afficher une liste des premiers nombres premiers
 Saisir un nombre puis afficher tous les nombres premiers inférieurs à ce nombre.

Soit un nombre n :
 →on teste avec une variable i tous les nombres de 2 jusqu'à n-1 et on affiche ceux qui sont premiers
 →on étudie le reste de la division euclidienne de i par tous les nombres allant de 2 à i-1

Programme python :

```
n=int(input("saisir un nombre"))
for i in range(2,n):
    P=0
    for j in range (2,i):
        if i%j==0:
            P=1
    if P==0:
        print(i)
```

```
from math import *
n=int(input("saisir un nombre : "))
if n>=2:
    print(2)
for i in range(3,n,2):
    P=0
    e=floor(sqrt(n))+1
    e=int(e)
    for j in range (2,e):
        if i%j==0:
            P=1
    if P==0:
        print(i)
```

→si n = 25,
 2
 3
 5
 7
 11
 13
 17
 19
 23

Exercice 5 : Afficher la liste des carrés des nombres impairs

Construire un algorithme puis un programme en python permettant d'afficher, pour un nombre donné, la liste des carrés des nombres impairs inférieurs à ce nombre.

Vous devrez utiliser au moins une fonction dans votre programmation.

On ne connaît pas le nombre de nombres à calculer donc on utilisera une boucle tant_que.

Algorithme :

Saisir un nombre n
Initialiser une variable i à 1.
Tant_que $i^2 < n$
 Afficher i^2
 Augmenter i de 2

Saisir un nombre n
 $i \leftarrow 1$.
Tant_que $i^2 < n$
 Afficher i^2
 Augmenter i de 2

Programme python :

```
def liste_carre_impairs(n):  
    i=1  
    while i**2 < n:  
        print(i**2)  
        i=i+2  
  
n=int(input("Veuillez saisir un nombre positif : "))  
if n>0:  
    liste_carre_impairs(n)  
else :  
    print("Le nombre saisi est négatif ou nul, veuillez relancer le programme")
```

Exemple : si on saisit 100 :

1
9
25
49
81