

tp5 algo

TP 5

Exercice 1 :

Programmer l'exercice 1 du TD 4. Rappel de l'énoncé :

Écrire un algorithme permettant d'afficher tous les nombres parfaits compris entre 1 et n, avec n saisi par l'utilisateur.

Rappel : un nombre est dit parfait lorsqu'il vaut la somme de tous ses diviseurs (lui-même exclu).

Exemple : 6 est un nombre parfait, car $6 = 1+2+3$

Présenter cet algorithme sous la forme d'une fonction ainsi qu'un programme l'utilisant

Exercice 2 :

Programmer l'exercice 2 du TD 4. Rappel de l'énoncé :

Écrire une fonction permettant de vérifier qu'une date est valide (déjà fait en TP) sachant que :

- on se place exclusivement dans le cadre du calendrier grégorien, à partir de l'année 1583 (les dates plus anciennes seront considérées comme non valides).

- aucun "si alors sinon" n'est autorisé dans la fonction

- la fonction doit être utilisable par le programme suivant :

```
programme verification_date
debut
...
fin verification_date
```

Exercice 3 :

Écrire un programme qui demande à l'utilisateur d'entrer une valeur de **n** strictement positive puis affiche le « triangle » suivant (les bords doivent être "droits") :

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
...
...
...
1 2 3 4 5 6 7 ... n
```

Exercice 4 :

Écrire un algorithme qui lit une valeur n entrée par l'utilisateur, puis affiche un carré de nombres de côté $2n + 1$, composé de la manière suivante (exemple donné avec $n = 4$) :

```
8 7 6 5 4 5 6 7 8
7 6 5 4 3 4 5 6 7
6 5 4 3 2 3 4 5 6
5 4 3 2 1 2 3 4 5
4 3 2 1 0 1 2 3 4
5 4 3 2 1 2 3 4 5
6 5 4 3 2 3 4 5 6
7 6 5 4 3 4 5 6 7
8 7 6 5 4 5 6 7 8
```

exercice 1

```
def is_perfect(number):  
    """Checks if a given number is perfect.  
  
    A perfect number is a positive integer that is equal to the sum of its proper divisors.  
  
    Args:  
        number: The number to check.  
  
    Returns:  
        True if the number is perfect, False otherwise.  
    """  
  
    if number <= 0:  
        return False  
  
    divisors_sum = 0  
    for divisor in range(1, number // 2 + 1):  
        if number % divisor == 0:  
            divisors_sum += divisor  
  
    return divisors_sum == number  
  
if __name__ == "__main__":  
    number = int(input("Enter a number to check: "))  
  
    while number <= 0:  
        number = int(input("Please enter a positive number: "))  
  
    for num in range(1, number + 1):  
        if is_perfect(num):  
            print(num, "is a perfect number")
```

exercice 2

```
def vérifier_date(jour:int,mois:int,annee:int):
    est_biss:bool
    mois_ok:bool
    jour_ok:bool
    fevrier_ok:bool
    date_ok :bool

    est_biss=((annee%4==0)and(annee%100!=0)) or (annee%400==0)
    mois_ok=((mois>=1) and (mois <=12))
    fevrier_ok=((est_biss)and (jour<=29)) or (est_biss==False) and (jour<=28)
    jour_ok= ((jour>=1) and (fevrier_ok or ((mois!=2 and mois != 4 and mois!=6 and mois!=9 and mois !=11) and jour<=31) or ((mois==4 or mois==6 or
    date_ok= jour_ok and mois_ok and (annee>1582)

    return date_ok

if __name__ == "__main__":
    day:int
    month:int
    year:int

    day=int(input("entrer un jour: "))
    month=int(input("entrer un mois: "))
    year=int(input("entrer une annee: "))

    if vérifier_date(day,month,year):
        print("la date est juste")
    else:
        print("la date est fausse")
```

```
((mois==4 or mois==6 or mois==9 or mois==11)and jour <=30)) )
```

ex 3

```
if __name__ == "__main__":
    rows: int

    rows=int(input("entrer nb de rangs"))

    for i in range (0, rows):
        for j in range (i+1):
            print(j+1, end="")
        print()
```

programme triangle_rectangle_sur_la_gauche

début

```
avec: rangs, compteur_i, compteur_j:entiers
afficher "saisir un nb de rangs"
saisir rangs
#vérifier saisie>0
```

```
pour compteur_i de 0 à rangs faire
    pour compteur_j de 0 à i+1 faire
        afficher (compteur_j+1)
    retour à la ligne
fin faire
```

fin triangle_rectangle_sur_la_gauche

Exercice 4

```
programme carré_2n+1_coté
début
    avec : nombre,nb,compteur_i, compteur_j, compteur_l: int

    afficher ("saisir un nombre: ")
    saisir nombre

    #check nombre>0

    nb← nombre*2

    pour compteur_i de (0 à nombre+1) faire
        pour compteur_j de (0 à nombre) faire
            afficher (nb)
            nb←nb+1
        fin faire
        pour compteur_l de 0 à nombre) faire
            afficher (nb)
            nb← nb-1
        fin faire
        afficher(nb)
        à la ligne
        nb←nb-1
    fin faire
    nb← nb+1
    pour compteur_i de (1 à nombre+1) faire
        nb←nb+1
        pour compteur_j de (0 à nombre) faire
            afficher (nb)
            nb←nb-1
        fin faire
        pour compteur_l de 0 à nombre) faire
            afficher (nb)
            nb← nb+1
        fin faire
        afficher(nb)
        à la ligne
    fin faire
fin carré_2n+1_coté
```

```

nombre=int(input("entrer nb de rangs: "))
while (nombre<=0):
    nombre=int(input("entrer nb de rangs: "))

nb=nombre*2

for i in range (0,nombre+1):
    for j in range (0,nombre):
        print(nb, end=" ")
        nb=nb-1
    for l in range (0,nombre):
        print(nb,end=" ")
        nb=nb+1
    print(nb)
    print()
    nb=nb-1
nb=nb+1
for i in range (1, nombre+1):
    nb=nb+1
    for j in range (0,nombre):
        print (nb,end=" ")
        nb=nb-1
    for l in range (0,nombre):
        print(nb,end=" ")
        nb=nb+1
    print(nb)
    print()

```