

Récupérer le fichier : « TS\_ISN\_TP10.zip » et le décompresser dans un dossier.

- TS\_ISN\_TP09\_Secu.py

1. Reprendre le programme TS\_ISN\_TP09\_Secu.py créé dans le TP précédent et le modifier en créant trois fonctions :

- *verif*(num) : qui affiche si le numéro est valide ou non.
- *n\_annee*(num) : qui renvoie l'année de naissance
- *n\_mois*(num) : qui renvoie le nom du mois de naissance

où num est le numéro de sécurité sociale.

2. Créer *table*(n, op, debut, fin) où :

- n : un paramètre entier
- op : un caractère "+" ou "\*".
- debut : un paramètre entier optionnel qui vaut 1 par défaut
- fin : un paramètre entier optionnel qui vaut 10 par défaut

```
Exemple :
table(5, "*", 3, 8)
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 25
5 * 8 = 30
```

Remarque : On pourra utiliser la commande *eval*()

3. Créer *factorielle*(n) qui renvoie la valeur de n ! pour un entier n.

Définition : Si  $n = 0$  ou si  $n = 1$ , alors  $n! = 1$ .  
 Si  $n \geq 2$  alors,  $n! = n \times (n - 1) \times \dots \times 2 \times 1$ .  
 C'est-à-dire que :  $n! = (n - 1)! \times n$ .

4. Créer *binom*(n, p) où n et p sont deux entiers qui renvoie la valeur du coefficient binomial  $\binom{n}{p}$ .

Rappels de 1°S : On a :  $\binom{n}{0} = 1$  et  $\binom{n}{n} = 1$

De plus, si  $p < n$  :  $\binom{n}{p} + \binom{n}{p+1} = \binom{n+1}{p+1}$

5. Reprendre la fonction *binom*(n, p) créée précédemment et y ajouter en 1<sup>ère</sup> ligne l'instruction : `print("binom(", n, ", ", p, ")")`

Lancer alors le calcul de *binom*(8, 5)

Que remarque-t-on ?

Pour éviter ce phénomène, on peut utiliser une mémoïsation des valeurs en utilisant un dictionnaire. Modifier alors la fonction *binom*(n, p) :

Ajouter en 1<sup>ère</sup> ligne l'instruction :

`print("binom_m(", n, ", ", p, ")")`

Lancer alors le calcul de *binom\_m*(8, 5)

Comparer avec le calcul sans mémoïsation.

```
def binom_m(n,p) :
    """mémoïsation"""
    global b_dico
    if (n,p) in b_dico :
        return b_dico[n, p]
    else :
        ...
        b_dico[n, p] = coef
    return coef

b_dico = {}
```

6. En utilisant la fonction *binom*(n, p) ou plutôt la version avec mémoïsation *binom\_m*(n, p), créer la fonction *pascal*(maxi) affiche les valeurs du triangle de Pascal jusqu'à une valeur maximale.

```
Exemple :
pascal(5)

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5
```