

Question 1: Ecrire une fonction

`interpretations(nbVar)` qui renvoie le tuple constitue de toutes les interpretations possible de nbvar variables propositionnelles

la technique que j'ai opte est de calculer tous les nombre possible en binaire jusqu'a 2^{nbvar} , puis de les retranscrire en tuple de vrai/faux. Voici le code (on assume une fonction translate to tuple defini comme le suit)

Q1: ecrire une fonction `Inter(nbvar)` qui renvoie le tuple constitue de toutes les interpretations possible de nbvar variables propositionnelles

```
def translate_totuple(binary: str):
    result = []
    for i in binary:
        if i == '1':
            result.append(True)
        else:
            result.append(False)

    return tuple(result)

def inter(nbvar):
    finalresult = ()
    for i in range(nbvar**2):
        result = bin(i)
        result = result[2:]
        while len(result) < nbvar:
            result = '0'+result
        result = translate_totuple(result)
        finalresult += result,
    return finalresult
```

Question 2.

Une formule propositionnelle FP de n variables est codee par une chaine de caracteres respectant la syntaxe python. les variables étant toujours codées `V[0]`, `V[1]`, ..., `V[n-1]`. Écrivez une fonction `TV(FP,n)` qui renvoie la table de vérité de la formule FP sous forme de tuple de tuples à l'aide de la fonction `Inter` et la fonction d'évaluation `eval(chaine)` du Python qui évalue une chaine de caractères si elle respecte la syntaxe du langage Python.

Exemple. Avec la chaîne de caractère `FP = "V[0] and V[1]"`, l'appel de la fonction `TV(FP,2)` doit renvoyer le tuple `((False,False,False), (False,True,False), (True,False,False), (True,True,True))`

le code est incomplet, mais le concept est juste. on genere les combinaison, puis on laisse exec evaluer l'expression, enfin on append dans la variable resultat final et on renvoie (le code n'est pas au point mais le concept est celui la)

```
def TV(fp, nbvar):  
    variables = inter(nbvar)  
    endresult = ()  
    for V in variables:  
        endresult += (i, (eval(fp)),)  
  
    return endresult
```

Question 3

Écrivez une fonction Satisfaisable(FP,n) qui renvoie la liste des interprétations des n variables propositionnelles qui satisfont la formule passée en paramètre. La question la plus facile, il suffit d'utiliser la fonction tv qu'on a précédemment crée, et récupérer le dernier élément de chaque tuple et vérifier si c'est vrai.

```
def Satisfaisable(fp, n):  
    verites = TV(fp, nbvar)  
    result = []  
    for i in verites:  
        if i[-1] == True:  
            result.append(i[:1])  
    return result
```