

Mathématiques Discrètes

feuille numéro 02

14 septembre 2018

Exercice 1

Q 1.1 De combien de manière un étudiant peut répondre à un questionnaire comportant dix questions où il faut répondre par oui ou par non ?

Q 1.2 On reprend le même test, mais sachant qu'il est prévu d'appliquer une pénalité en cas de mauvaise réponse, on laisse la possibilité de ne pas répondre à une question pour éviter de recevoir la pénalité.

Q 1.3 Reprendre les questions précédentes sachant que le test est maintenant un questionnaire à choix multiple où il y a quatre propositions (et une seule est exacte).

Q 1.4 Reprendre la première question sachant que le test est maintenant un questionnaire à choix multiple où il y a quatre propositions et tous les cas de figures sont possibles (0 bonne réponse, 1 bonne réponse ... 4 bonnes réponses).

Exercice 2

Q 2.1 De combien de manières sept personnes peuvent être installées sur une table ronde ? (On considère comme identiques tout couple d'arrangements dont l'un peut être obtenu en faisant une rotation de l'autre.)

Q 2.2 Si deux individus insistent pour être côte à côte, combien reste-t-il de possibilités ?

Exercice 3

Trouvez les valeurs de n pour chacune des équations suivantes :

Q 3.1 $A_n^2 = 90$

Q 3.2 $A_n^3 = 3A_n^2$

Q 3.3 $2A_n^2 + 50 = A_{2n}^2$

Exercice 4

On donne le code suivant.

```
def mystere(n):
    r=set()
    for i in range(n):
        for j in range(n):
            if i != j:
                r.add( (i,j) )
    return r
```

Q 4.1 Spécifiez la fonction `mystere`.

Q 4.2 Donnez en fonction de n la valeur de l'expression `len(mystere(n))`.

Q 4.3 Combien y a-t'il de couples d'entiers naturels distincts, dont les deux composantes sont inférieures strictement à n et dont la première composante est strictement inférieure à la seconde ?

Q 4.4 En déduire une expression en fonction de n de $\sum_{i=0}^{i=n-1} i$

Exercice 5

On donne le code suivant :

```

def existe_doublon_naivement(liste):
    """ Teste si une liste contient des doublons
    CU: liste est une liste de chaines de
    caractere
    ExU:
    >>> existe_doublon_naivement(["tata","titi","toto","titi"])
    True
    >>> existe_doublon_naivement(["tata","titi","toto","tete","tyty"])
    False
    """
    res=False
    n=len(liste)
    for i in range(n-1):
        for j in range(i+1,n):
            if liste[i]==liste[j]:
                res=True
    return res

```

Q 5.1 Combien y a t'il de test d'égalités entre chaine de caractères lors de l'appel avec un paramètre liste de longueur n ?

Exercice 6

Q 6.1 implantez en python la fonction factorielle

Exercice 7

Q 7.1 implantez en python la fonction arrangement