

NSI Amérique du Nord 2024 Jour 2

EXERCICE 1

- 1) def echange(tab,i,j) :
 tampon=tab[i]
 tab[j]=tab[i]
 tab[i]=tampon
- 2) ligne 6 : triStooge(tab,i,j-k)
 ligne 7 : triStooge(tab,i+k,j)
 ligne 8: triStooge(tab,i,j-k)
- 3) récursif car la fonction est appelée en son sein.
- 4) $k=(5-0+1)//3=6//3=2$
- 5) $9 \times 3 + 3 \times 3 + 3 = 3 \times 13 = 39$
- 6) 1 : triStooge(A,1,3)
 2 : triStooge(A,2,3)
 3 : triStooge(A,0,3)
- 7) ???
- 8) Tri fusion

EXERCICE 2

- 1) Dufour Marc
 Martin Sophie
- 2) SELECT nom_medic FROM medicament
 WHERE prix < 3;
- 3) INSERT INTO client (id_client,nom_client,prenom_client,num_secu_sociale)
 VALUES (3,'DURAND','Nathalie',269054958815780)
- 4) Id_client et id_medic pour lier la table ordonnance aux tables client et
 medicament
- 5) Ligne 7 : 1
 Ligne 8 : 3
- 6) UPDATE medicament SET quantite=447 WHERE nom_medic='Acide
 ascorbique';
- 7) $2 \times 2,30 + 3,50 + 3 \times 5,50 = 4,6 + 19 = 23,60$ Euros
- 8) SELECT medicament.nom_medic FROM medicament
 JOIN ordonnance ON ordonnance.id_medic=medicament.id_medic
 WHERE ordonnance.id_ordo=6

EXERCICE 3

- 1) Charlie : 192.168.1.3 elle doit commencer par 192.168.1 à cause du masque de
 sous-réseau et pour le dernier octet, 0,1,2 et 155 sont déjà pris.
- 2) [Transaction('Alice','Bob',10),Transaction('Bob','Alice',5)]
- 3) Bloc0 est le premier bloc créé avec la class Blockchain qui crée un bloc dont le
 précédent est None. (ligne 20)
- 4) Pour le bloc1 , on a bloc-precedent = Blockchain()
- 5) ma_blockchain=Blockchain()

- ```

mablockchain.tete=Bloc([Transaction('Alice','Charlie',50),...,...],mablockchain.tete
)
mablockchain.tete=Bloc([Transaction('Bob','Charlie',20),...,...],mablockchain.tete
)

```
- 6) Bob :  $100 + 30 - 20 - 20 = 90$
  - 7) Def ajouter\_bloc(self,liste\_transactions) :
 

```

 self.tete=Bloc(liste_transactions,self.tete)

```
  - 8) Liste de diffusion ou Broadcast 192.168.1.255
  - 9) Ligne 5 : `solde=self.precedent.calculer_solde(utilisateur)`  
 Ligne 7 : `if transaction[0] == utilisateur`  
 Ligne 8 : `solde=solde – transaction [2]`  
 Ligne 9 : `elif transaction[1] == utilisateur`  
 Ligne 10 : `solde=solde – transaction[2]`
  - 10) `Ma_blockchain.calculer_solde('Alice')`
  - 11) Cela consiste à tester toutes les valeurs possibles de nonce pour trouver celle qui convient
  - 12) Pour bloc0, `hash_bloc_precedent=0` (ligne 15) car le bloc precedent de bloc0 est None
  - 13)  $2^{256}$
  - 14) Ligne 5 : `while self.hash[:2]!='00' :`  
 Ligne 6 : `self.nonce=self.nonce+1`  
 Ligne 7 : `self.hash=self.calculer_hash()`