



QCM Liste Chainées

▼ Question 1

```
void supprimer(Liste *liste, int pos)
{
    if (pos < 0 || pos > liste->compteur-1)
    {
        cout << "Erreur! La position est invalide." << endl;
        return;
    }
    if (pos == 0)
    {
        Noeud *del = liste->entete;
        liste->entete = liste->entete->suivant;
        delete del;
    }
    else
    {
        Noeud *courant = liste->entete;
        for (int i = 0; i < pos-1; i++)
        {
            courant = courant->suivant;
        }
        Noeud del = courant->suivant;
        courant->suivant = courant->suivant->suivant;
        delete del;
    }
    liste->compteur--;
}
```

▼ Question 2

Laquelle des structures de données suivantes a ses données contiguës en mémoire ?

-un tableau

▼ Question 3

Dans une liste chaînée, comment fait-on pour accéder au dernier élément de la liste ?

-on parcourt tout les éléments du premier au dernier, dont le pointeur vers l'élément suivant vaut nullptr

▼ Question 4

```
void ajouterEnTete(List * liste, int valeur)
{
    Noeud * nouveauNoeud = new Noeud;
    if (nouveauNoeud == nullptr)
    {
        exit(EXIT_FAILURE);
    }
    nouveauNoeud->donnees = valeur;
    nouveau->suivant = liste->entete;
    liste->entete = nouveau;
    liste->compteur++;
}
```

▼ Question 5

Dans une liste chaînée, chaque noeud contient au moins deux champs. L'un des deux champs contient l'élément, que contient le deuxième champ ?

-un pointeur sur un noeud

▼ Question 6

Erreur définition structure Noeud

```
Noeud *suivant=nullptr;
```

▼ Question 7

Méthode LIFO

-la pile

▼ Question 8

Les listes chaînées sont meilleures...

pour les structures dont la taille et les données sont en constante évolution

▼ Question 9

quelle est la structure de données qui permet la suppression à une extrémité de la liste et l'insertion à l'autre extrémité

la liste doublement chaînée

▼ Question 10

avantages liste chaînée par rapport à un tableau

une liste chaînée n'a pas de taille fixe