Pointeurs Intelligents

Question 1 :

1. Lorsque le décompte de références d'un objet géré atteint _____, le destructeur de l'objet géré est invoqué.

zero

Question 2 :

2. Les pointeurs faibles ne partagent pas _____ de l'objet géré.

la possesion

Question 3 :

3. Que signifie l'acronyme `RAII` ?

L'acquisition de ressources est une initialisation

Question 4 :

4. Les pointeurs intelligents peuvent aider à prévenir lequel des événements suivants ?

Fuites de mémoire(Memory leaks) - Pointeurs non initialisés(Unintialized pointerts) - Pointeurs pendants(Dangling pointers)

Question 5 :

5. Les pointeurs uniques ne peuvent être que _____.

déplacé

Question 6 :

6. La méthode `use_count()` renvoie le _____.

le nombre de références du pointeur

Question 7 :

7. Supposons que nous ayons un nœud dans une liste à double lien défini comme suit avec des pointeurs bruts. Quel pointeur intelligent devrions-nous utiliser à la place des pointeurs bruts ?

```
class Node {
   int data;
   Node *next;
   Node *prev;
};
```

shared_ptr et weak_ptr

Question 8 :

8. Nous pouvons fournir des pointeurs intelligents avec des fonctions _____ personnalisées qui seront appelées pour gérer la destruction d'objets.

deleter

Question 9 :

9. La méthode `get()` pour les pointeurs intelligents renvoie _____.

Le pointeur brut en cours de gestion.

Question 10 :

10. Lorsque vous déclarez un objet pointeur intelligent, cet objet est placé sur le _____.

stack