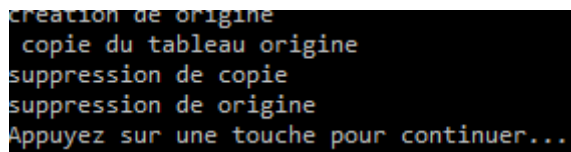


Exercice classe canonique copy/move

- 1) Ecrire une classe Tableau telle que

```
Tableau tab{ 42,"origine" };
```

```
Tableau tab_copy = tab;
```



```
creation de origine  
copie du tableau origine  
suppression de copie  
suppression de origine  
Appuyez sur une touche pour continuer...
```

- 2) Mettre en place l'opérateur d'affectation de copie

Constatez que le code suivant pose un problème :

```
Tableau tab_autre{ 32,"autre" };  
tab_copy = tab_autre;
```

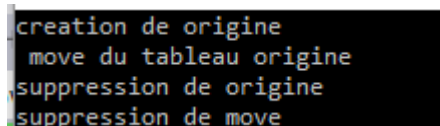
Corrigez en créant l'opérateur d'affectation de copie.

- 3) Mettre en place le constructeur de move pour que :

Pour provoquer la création d'un objet temporaire, créer l'objet dans la fonction fabrique, puis le retourner (ne pas fusionner la création et le return, car sinon le compilateur optimise la création de l'objet)

```
Tableau tab_copy = fabrique("origine");
```

Résultat à obtenir :



```
creation de origine  
move du tableau origine  
suppression de origine  
suppression de move
```

Constatez la compilation si vous supprimez l'opérateur de copie que VOUS avez écrit.

- 4) Mettre en place l'opérateur d'affectation de move

Pour l'utiliser, provoquer la création d'un objet temporaire.

5) En utilisant une classe comme :

```
struct Qualif {  
    std::string data;  
    Qualif(std::string nom) : data(nom) {  
        std::cout << " creation de qualif " << data << std::endl;  
    }  
  
    Qualif(const Qualif& other): data(other.data){  
        std::cout << " cpy ctor de qualif " << data << std::endl;  
    }  
  
    Qualif& operator=(const Qualif& other) {  
        data = other.data;  
    }  
};
```

Et en modifiant Tableau :

```
class Tableau  
{  
private:  
    int *_p_vals;  
    int _size;  
    Qualif _nom;  
  
public:
```

Faire en sorte que le move du Tableau provoque le move de Qualif, en utilisant std::move.