TP4 – notions avancées

Exercice 1 : les templates

Créez une fonction "trier" pour trier un tableau de 10 données avec un tri par sélection. Le tableau devra être de n'importe quel type. Cette fonction utilisera une autre fonction, "échanger", pour échanger les éléments du tableau.

Exercice 2 : les pointeurs sur fonctions

Etant donner un tableau de valeurs entières, on veut créer des filtres pour le contenu de ce tableau. Un filtre est une manière de traiter les données du tableau.

On peut :

Afficher les valeurs positives

Afficher les valeurs négatives

Afficher les valeurs supérieures à un seuil

On se propose d'écrire les fonctions précédentes pour chaque entrée du tableau (exemple : la fonction afficherPositif(int a), affiche a s'il est positif).

On définira une fonction « appliquerFiltre » qui prend en paramètre le tableau et le filtre à appliquer (pointeur sur la fonction à appliquer).

On écrira une fonction principale main qui créer le tableau et affichera un menu à l'utilisateur pour choisir quel filtre à appliquer.

Proposition de correction :

```
#include <iostream>
#include<vector>

void afficherP(int& x, int seuil = 0) {
	if (x >= 0)
		std::cout << x << "\t";
}
void afficherN(int& x,int seuil = 0) {
	if (x < 0)
		std::cout << x << "\t";
}
void greaterThan(int& x, int seuil)
{
	if(x>=seuil)
		std::cout << x << "\t";
}
void appliquerFiltre(std::vector<int>& vec, void(*filtre)(int&,int))
{
	int seuil=0;
	if (filtre == greaterThan) {
		std::cout << "Donner seuil:";
		std::cin >> seuil;
	}
	for (int& e : vec) {
		filtre(e,seuil);
```

```
	}
}

int main()
{
	std::vector<int> tab{ -2,3,-33,7,99,6,2,-39 };
	std::cout << "************************\n";
	std::cout << "* 1. Afficher positif  *\n";
	std::cout << "* 2. Afficher Negatif  *\n";
	std::cout << "* 3. Superieur A       *\n";
	std::cout << "************************\n";
	std::cout << "Entrez votre choix?\n";
	int choix;
	std::cin >> choix;
	switch (choix)
	{
	case 1: appliquerFiltre(tab, afficherP);
		break;
	case 2: appliquerFiltre(tab, afficherN);
		break;
	case 3:
		appliquerFiltre(tab, greaterThan);
	default:
		break;
	}
	return 0;
}
```

Exercice 3 : les fonctions lambdas

Reprendre l'exercice précédent en utilisant les fonctions lambdas

Exercice 4 : paramètres par défaut d'une fonction

On veut écrire une fonction ***calculerPrix*** qui permet à un commerçant d'avoir le prix final d'un produit qui peut être soumis à une taxe et à une remise.

Tout produit est taxé à 19% sauf les produits de luxe qui sont taxés à 25%.

Un produit peut être mis en solde (20%) ou non.

1. Définir la structure produit sachant qu'un produit est caractérisé par son type, son prix hors taxe et un booléen indiquant s'il est soldé.
2. Ecrire la fonction calculerPrix qui retourne le prix TTC net (après un éventuel solde)
3. Ecrire la fonction main et y créer des produits et afficher pour chacun son prix net.