IF122 : Correction du contrôle intermédiaire

Exercice 1 :

```
Tableau concatener(Tableau t1, Tableau t2) {
    Tableau r = creer_tableau(t1.size+t2.size);
    int i;
    for (i=0;i<t1.size;i++) {
        r.contenu[i] = t1.contenu[i];
    }
    for (i=0;i<t2.size;i++) {
        r.contenu[t1.size+i] = t2.contenu[i];
    }
    return r;
}
```

Exercice 2 :

```
static int v = 1;       : globale, X, statique
int w = 0;              : globale, X, non statique

int f(int v) {          : argument, valeur, X
  v=v+1;
  return v;
}
int g() {
  v++;
  return v-1;
}
int h(int &v) {         : argument, référence, X
  v = v - w;
  return v;
}
int i(int w) {          : argument, valeur, X
  static int v = 10;    : locale, X, statique
  w = v+1;
  return v;
}
int j(int v) {          : argument, valeur, X
  return f(v)+1;
}
```

Exercice 3 : L'affichage nous donne : 1, 5, 2, 1, 10, 7, 2, 10

Exercice 4 :

- $\#include < iostream >$: inclusion de la bibliothèque définissant cin et cout ;
- $int\ main()\{$: déclaration de la fonction principale du programme ;
- $int\ *pe1...$: déclaration d'une variable locale pe1 en tant que pointeur vers un entier ;
- $... = new\ int[5]$: réservation dynamique de 5 cases mémoires consécutives stockant des entiers. pe1 pointe vers la première de ces 5 cases.
- $int\ **pe2...$: déclaration d'une variable locale pe2 en tant que pointeur vers un pointeur d'entier ;

- $... = new\ (int*)[6]$: réservation dynamique de 6 cases mémoires consécutives stockant des "pointeurs d'entier". pe2 pointe vers la première de ces 6 cases.
- $pe2[0] = pe1$: affectation de pe1 à la première case de pe2. pe2 est un pointeur vers un tableau de 6 cases de "pointeur d'entier". La case 0 est donc de type "pointeur d'entier" et pe1 est un pointeur d'entier. L'affectation est donc autorisée.
- $for(int\ i = 0; i < 6; i++)$: boucle faisant varier l'indice i de 0 inclus jusqu'à 6 exclus. L'indice i est défini uniquement à l'intérieur de cette boucle ;
- $cin >> pe1[i]$: demande une valeur à l'utilisateur et la stocke dans la case i de pe1. pe1 est un tableau d'entier donc la valeur attendue est un entier (pas de problème pour cin). Par contre pe1 est associé à un tableau de 5 cases donc la case d'indice 5 n'existe pas. La boucle doit être corrigée pour faire varier i de 0 à 5 exclus.
- $for(int\ i = 0; i < 6; i++)$: boucle faisant varier l'indice i de 0 inclus jusqu'à 6 exclus.
- $cout << pe2[0][i]$: pe2 est un pointeur vers un pointeur d'entier. Il pointe vers un tableau de 6 cases. $pe2[0]$ accède à la première case de ce tableau. La première case de ce tableau est pe1, qui est un tableau de 5 entiers. $pe2[0][i]$ accède donc à la i-ème case de pe1. Nous affichons donc à l'écran les valeurs entrées précédemment par l'utilisateur. Nous avons encore une erreur dans l'indice de la boucle qui doit s'arrêter à 5 et non 6.
- Le programme principal se termine mais les deux zones mémoires n'ont pas été libérées par un delete ;
- La fonction main est déclarée comme retournant un entier mais il n'y a aucun return dans la fonction.

Exercice 5 :

```
t_list cons_sorted(t_element e, t_list l) {
    if (isEmpty(l)) {
        return cons(e,l);
    } else if (!lower_element(head(l), e)) {
        return cons(e,l);
    } else {
        return cons(head(l), cons_sorted(e, queue(l)));
    }
}
```

Exercice 6 :

```
t_list search_sorted(t_element e, t_list l) {
    if (isEmpty(l)) {
        return EMPTY_LIST;
    } else if (lower_element(e, head(l))) {
        return EMPTY_LIST;
    } else if (equal_element(e, head(l))) {
        return l;
    } else {
        return search_sorted(e, queue(l));
    }
}
```