INFO-H-100 - Programmation
TP 11 - Les tris
Corrections

Exercices 11.1 à 11.3

```
#include <iostream>
using namespace std;

const int MAX = 10;
typedef int Element;
typedef Element Vecteur[MAX];
void print(Vecteur, int);
bool compare(Element, Element);
void swap(Element&, Element&);
void triSelection(Vecteur, int);   //Ex 11.1
void triInsertion(Vecteur, int);   //Ex 11.2
void triBulle(Vecteur, int);       //Ex 11.3

int main()
{
  //Ex 11.1
  Vecteur v1 = {10, 4, -5, 1, 11, 0, 7, 2, 1, -1};
  print(v1, 10);
  triSelection(v1, 10);
  print(v1, 10);

  //Ex 11.2
  Vecteur v2 = {10, 4, -5, 1, 11, 0, 7, 2, 1, -1};
  print(v2, 10);
  triInsertion(v2, 10);
  print(v2, 10);

  //Ex 11.3
  Vecteur v3 = {10, 4, -5, 1, 11, 0, 7, 2, 1, -1};
  print(v3, 10);
  triBulle(v3, 10);
  print(v3, 10);

  return 0;
}


void print(Vecteur v, int n)
{
  for (int i = 0; i < n ; ++i)
    cout << v[i] << " ";
  cout << endl;
}
```

```
bool compare(Element a, Element b)
{
  return a < b;
}

void swap(Element& a, Element& b)
{
  Element tmp = a;
  a = b;
  b = tmp;
}

void triSelection(Vecteur v, int n)
{
  for (int i = 0 ; i < n-1 ; ++i)
  {
    int min_idx = i;
    for (int j = i+1 ; j < n ; ++j)
    {
      if (compare(v[j], v[min_idx]))
          min_idx = j;
    }
    swap(v[min_idx], v[i]);
  }
}

void triInsertion(Vecteur v, int n)
{
  for (int i = 1 ; i < n ; ++i)
  {
    int j;
    Element tmp = v[i];
    for (j = i-1 ;  (j >= 0) && compare(tmp, v[j]) ; --j)
      v[j+1] = v[j];
    v[j+1] = tmp;
  }
}

void triBulle(Vecteur v, int n)
{
  for (int i = 0 ; i < n - 1 ; ++i)
  {
    for (int j = n - 1 ; j > i ; --j)
    {
      if (compare(v[j], v[j-1]))
        swap(v[j-1], v[j]);
    }
  }
}
```

Exercice 11.4

```
#include <iostream>
using namespace std;

const int MAX= 100;
const int MAX_STRING = 128;
typedef char String[MAX_STRING];
typedef String StringArray[MAX];
void printString(StringArray, int);
void copy(String, String);
bool sontTries(String, String);
void triAlphabetique(StringArray, int);

int main()
{
  StringArray v = { "Christophe", "Aurelie",
                    "Frederic", "Boris",
                    "informatique", "polytechnique",
                    "ordinateur", "langage",
                    "programmation", "compilation",
                    "execution" };
  printString(v, 11);
  cout << endl;
  triAlphabetique(v, 11);
  printString(v, 11);

  return 0;
}

void printString(StringArray v, int n)
{
  for (int i = 0; i < n ; ++i)
    cout << v[i] << endl;
}

bool sontTries(String a, String b)
{
  //tri selon la valeur ascii (les majuscules sont avant les minuscules)

  int i;
  for (i = 0 ; a[i] != '\0' && b[i] != '\0' && a[i] == b[i] ; ++i);
  return a[i] <= b[i];
}

void copy(String a, String b)
{
  for (int i = 0 ; i < MAX_STRING ; ++i)
    b[i] = a[i];
}

void triAlphabetique(StringArray v, int n)
{
  for (int i = 0; i < n ; ++i)
  {
    int j;
    String tmp = "";
    copy(v[i], tmp);
    for (j = i-1 ; (j >= 0) && sontTries(tmp, v[j]) ; --j)
    {
      copy(v[j], v[j+1]);
    }
    copy(tmp, v[j+1]);
  }
}
```

Exercice 11.5

```
#include <iostream>
using namespace std;

const int MAX = 10;
const int MAX_LINE = 10;
const int MAX_COLUMN = 5;
typedef int Line[MAX_COLUMN];
typedef Line Matrix[MAX_LINE];
bool compare(Line, Line);
void copy(Line, Line);
void printMatrix(Matrix, int);
void triMatrice(Matrix, int);

int main()
{
  Matrix m = { {1, 4, 5, 2, 0}, { -1, 5, 2, 4, 1} };
  printMatrix(m, 2);
  cout << endl;
  triMatrice(m, 2);
  printMatrix(m, 2);

  return 0;
}

void print(Line l, int n)
{
  for (int i = 0; i < n ; ++i)
    cout << l[i] << " ";
  cout << endl;
}

bool compare(Line a, Line b)
{
  return a[0] < b[0];
}

void swap(Line a, Line b)
```

```
{
  for (int i = 0 ; i < MAX_COLUMN ; ++i)
    swap(a[i], b[i]);
}

void printMatrix(Matrix m, int n)
{
  for (int i = 0 ; i < n ; ++i)
  {
    print(m[i], MAX_COLUMN);
  }
}

void triMatrice(Matrix m, int n)
{
  for (int i = 0 ; i < n - 1 ; ++i)
  {
    for (int j = n - 1 ; j > i ; --j)
    {
      if (compare(m[j], m[j-1]))
        swap(m[j-1], m[j]);
    }
  }
}
```