

INFO-H-100 - Programmation
TP 12 - Exercices de synthèse
Corrections

Juin 2003 - Question 2

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

const int MAX = 128;
typedef char String[MAX];
bool compare(char, char, String);
void swap(char&, char&);
int strlen(String);
void tri_alpha(String, String);

int main()
{
    String O = "F+KB-AGRPZUM";
    String V = "A+F--+KABG";
    cout << O << endl;
    cout << V << endl;
    tri_alpha(V, O);
    cout << V << endl;

    return 0;
}

bool compare(char a, char b, String O)
{
    int i;
    for (i = 0 ; O[i] != '\0' && i < 30 && a != O[i] && b != O[i] ; ++i);
    return (a == O[i] && b != O[i]);
}

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

int strlen(String s)
{
    int len;
    for (len = 0 ; s[len] != '\0' ; ++len);
    return len;
}
```

```

void tri_alpha(String V, String O)
{
    int n = strlen(V);
    for (int i = 0 ; i < n -1 ; ++i)
    {
        for (int j = n - 1 ; j > i ; --j)
        {
            if (compare(V[j], V[j-1], O))
                swap(V[j], V[j-1]);
        }
    }
}

```

Jun 2007 - Question 3

```

#include <iostream>
using namespace std;

const int MAX_N = 100;
const int MAX_POLY = 5;
typedef int Polynome[MAX_N+1];
typedef Polynome PolyArray[MAX_POLY];
bool compare(Polynome, Polynome);
void copy(Polynome, Polynome);
void swap(int&, int&);
void printPolynome(Polynome);
void printPolyArray(PolyArray, int);
void sortPolynome(PolyArray, int);

int main()
{
    PolyArray p = { {1, 0, 2}, {5, 1, 0, 2, 1, 6, -1}, {0, 10}, {0, 1} };
    printPolyArray(p, 4);
    sortPolynome(p, 4);
    printPolyArray(p, 4);

    return 0;
}

bool compare(Polynome a, Polynome b)
{
    return (a[0] < b[0] || (a[0] == b[0] && a[a[0]+1] < b[a[0]+1]));
}

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

```

```

void copy(Polynome a, Polynome b)
{
    int max = (a[0] > b[0] ? a[0] : b[0]) + 1;
    for (int i = 0 ; i <= max ; ++i)
        b[i] = a[i];
}

void printPolynome(Polynome p)
{
    for (int i = 1 ; i <= p[0] ; ++i)
        if(p[i] != 0)
            cout << p[i] << " x^" << (i-1) << " + ";
    cout << p[p[0]+1] << " x^" << p[0] << endl;
}

void printPolyArray(PolyArray v, int n)
{
    for (int i = 0 ; i < n ; ++i)
        printPolynome(v[i]);
}

void sortPolynome(PolyArray v, int n)
{
    for (int i = 1 ; i < n ; ++i)
    {
        int j;
        Polynome tmp;
        copy(v[i], tmp);
        for (j = i-1 ; (j >= 0) && compare(tmp, v[j]) > 0 ; --j)
        {
            copy(v[j], v[j+1]);
        }
        copy(tmp, v[j+1]);
    }
}

```

Aout 2007 - Question 3

Voir exercice 11.4.