

```

/*****
IPV6.h
*****/
#pragma once
#include <string>
#include <vector>
using namespace std;

class IPV6
{
protected:
    string adresse;
    vector<string> strHextets;
    bool error;
    int compteur1;
    int position;

    void suppressionDesZerosDeDebut();
    void remplacementSuiteDeZeros();
    void reconstructionFinale();

    static vector<string> split(const string& s, char c);

public:
    IPV6(string adr);

    string getAdresse() { return adresse; }
    bool getError() { return error; }
};

```

Il faut commenter votre programme pour qu'il soit compréhensible !

```

/*****
IPV6.cpp
*****/
#include<iostream>
#include <string>
#include <vector>
#include "IPV6.h"

using namespace std;

IPV6::IPV6(string adr) {
    this->adresse = adr;
    this->strHextets = IPV6::split(this->adresse, ':');
    this->error = false;
    this->compteur1 = 0;
    this->position = 0;

    if (this->strHextets.size() == 8) {
        for (unsigned i = 0; i < this->strHextets.size(); i++) {
            if (strHextets[i].size() != 4)
                error = true;
        }
    }
    else
        error = true;
}

void IPV6::suppressionDesZerosDeDebut() {
    for (unsigned int j = 0; j < this->strHextets.size(); j++) {
        for (unsigned int k = 0; k < this->strHextets[j].size(); k++) {
            int recherche = true;
            while (recherche == true) {
                if (this->strHextets[j][k] == '0' && this->strHextets[j].size() - 1) {
                    this->strHextets[j].erase(k, k+1);
                }
                else
                    recherche = false;
            }
        }
    }
}

void IPV6::remplacementSuiteDeZeros() {
    int z, compteur2=0;
    string zero = "0";
    for (z = 0; z < this->strHextets.size(); z++) {
        if (this->strHextets[z].compare(zero)) {
            compteur1++;
            if (this->compteur1 >= compteur2) {
                compteur2 = compteur1;
            }
        }
    }
}

```

Votre constructeur doit appeler les méthodes qui simplifient l'adresse

Fonctionne

Ne fonctionne pas

```

        }
        }
    }
}

void IPV6::reconstructionFinale() {
    suppressionDesZerosDeDebut();
    remplacementSuiteDeZeros();
    this->strHextets.erase(this->strHextets.begin() + this->position, this->strHextets.begin() + this->position + this->compteur1);
    this->adresse = this->strHextets[0];
    for (int i = 1; i < this->strHextets.size(); i++)
    {
        if (this->position != i) {
            this->adresse += ":" + this->strHextets[i];
        }
        else
            adresse += this->strHextets[i];
    }
}

vector<string> IPV6::split(const string& s, char c)
{
    vector<string> v;
    unsigned int i = 0;
    unsigned int j = s.find(c);
    while (j < s.length()) {
        v.push_back(s.substr(i, j - i));
        i = ++j;
        j = s.find(c, j);
        if (j >= s.length()) {
            v.push_back(s.substr(i, s.length()));
            break;
        }
    }
    return v;
}

/*****
mainTest.cpp
*****/
#include <iostream>
#include <string>
#include <vector>
#include "IPV6.h"
using namespace std;

int main()
{
    string adresse = "2001:0db8:0000:85a3:0000:0000:ac1f:8001";
    IPV6 adr(adresse);

    //cout << "Saisir l'adresse IPV6 : " << endl;
    //getline(cin, adresse);

    if (adr.getError())
        cout << "Mauvaise adresse" << endl;
    else {
        cout << "Bonne adresse" << endl;
        cout << "Voici l'adresse IPV6 simplifie : " << endl;
        cout << adr.getAdresse() << endl;
    }

    cin.get(); cin.ignore();
    return EXIT_SUCCESS;
}

```

Cette ligne plante !

Ne fonctionne pas si la plus longue suite de zéros est au début ou à la fin