

```

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

class IPV6
{
private:
    string adresse;
    vector <string> strHextets;
    bool error;

    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; }
};

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

IPV6::IPV6(string adr)
{
    if (adresse.size() != 39 || adresse.size() > 'G' || 'g' || strHextets.size() > 4)
        this->error = getError();
}

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;
}

void IPV6::suppressionDesZerosDeDebut()
{
    int i, j;
    for (i = 0; i < adresse.size(); i++)
    {
        for (j = 0; j < strHextets.size(); j += 4)
        {
            ;
        }
    }
}

void IPV6::remplacementSuiteDeZeros()
{
    int i, j;
    for (i = 0; i < adresse.size(); i++)
    {
        for (j = 0; j < strHextets.size(); j += 4)
        {
            if (strHextets.begin() = 0)
            {
                ;
            }
        }
    }
}

```

Très insuffisant.

Il fallait au moins essayer de faire une méthode. Pour ça, il faut écrire des commentaires qui doivent détailler les actions à réaliser !

39 ???

Oups !

Oups !

```

    }
}

void IPV6::reconstructionFinale()
{
    int i, j;
    for (i = 0; i < adresse.size(); i++)
    {
        for (j = 0; j < strHextets.size(); j += 4)
        {
            ; Oups !
        }
    }
}

void affichage(const vector<string>& leVecteur)
{
    unsigned i;
    for (i = 0; i < leVecteur.size(); i++)
        cout << leVecteur[i];
}

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

using namespace std;

int main()
{
    vector<string> address;
    string Ladresse;

    cout << "Saisir l'adresse IPV6: ";
    getline(cin, Ladresse);
    address.push_back(Ladresse);
    cin.seekg(0, ios::end); cin.clear();

    cout << "L'adresse saisie est: ";
    affichage(address);

    cin.ignore();
    return EXIT_SUCCESS;
}

```