Coding game test

The Challenge : The goal is to find the maximum speed of a car to cross all the green lights without having to stop or slow down. The vehicle enters the zone directly at the programmed speed and its speed does not change afterwards.

The input : Maximum speed , LightCount (number of lights), distance and duration data.

The output : The maximum speed to solve the problem.

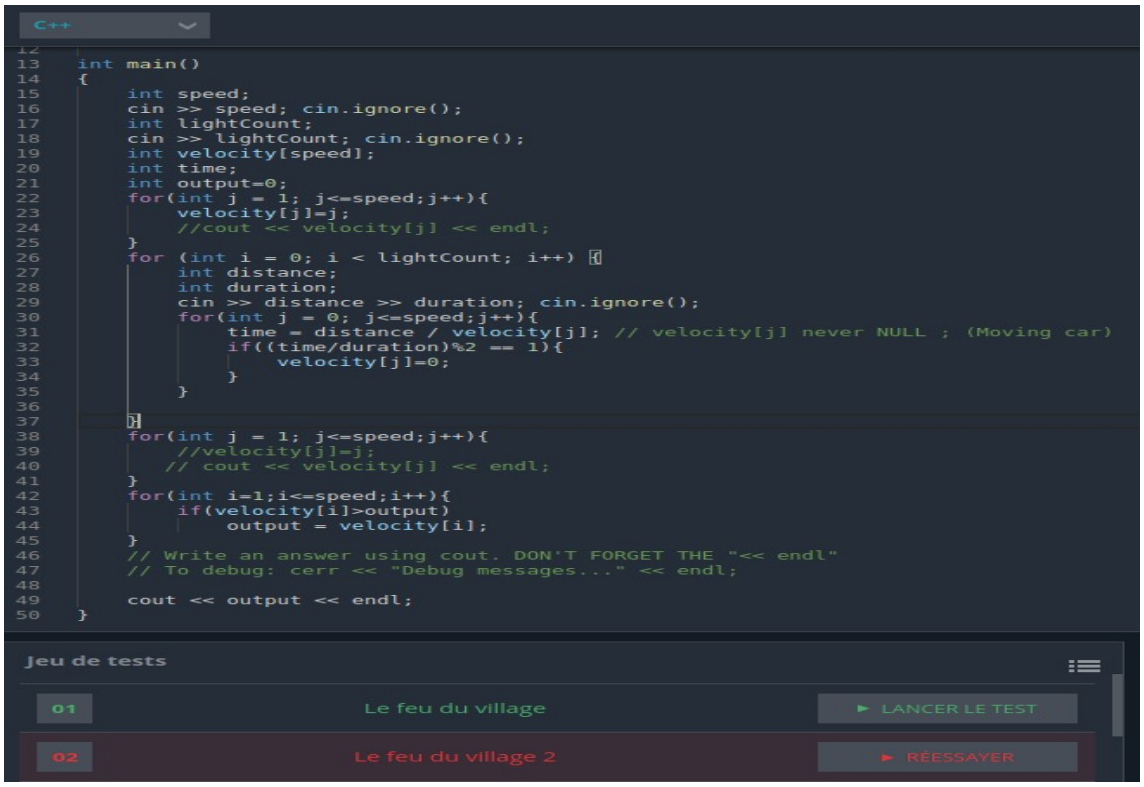
The Solution : I have first tried to find the optimum speed using math equations. However, I have noticed that it is not that complicated since the output is an integer and takes a determined number of values [0 : maximum speed allowed]. I have decided to check every single speed in that interval.

To arrive at the next zone at time (t1) (red light has just turned on to green), the time of the journey has to pass the next test:

(t1/duration)%2 = 0

If not, the car will be forced to stop since the next light will be red. (see example 3 below)

Coding : The first try is as shown below :



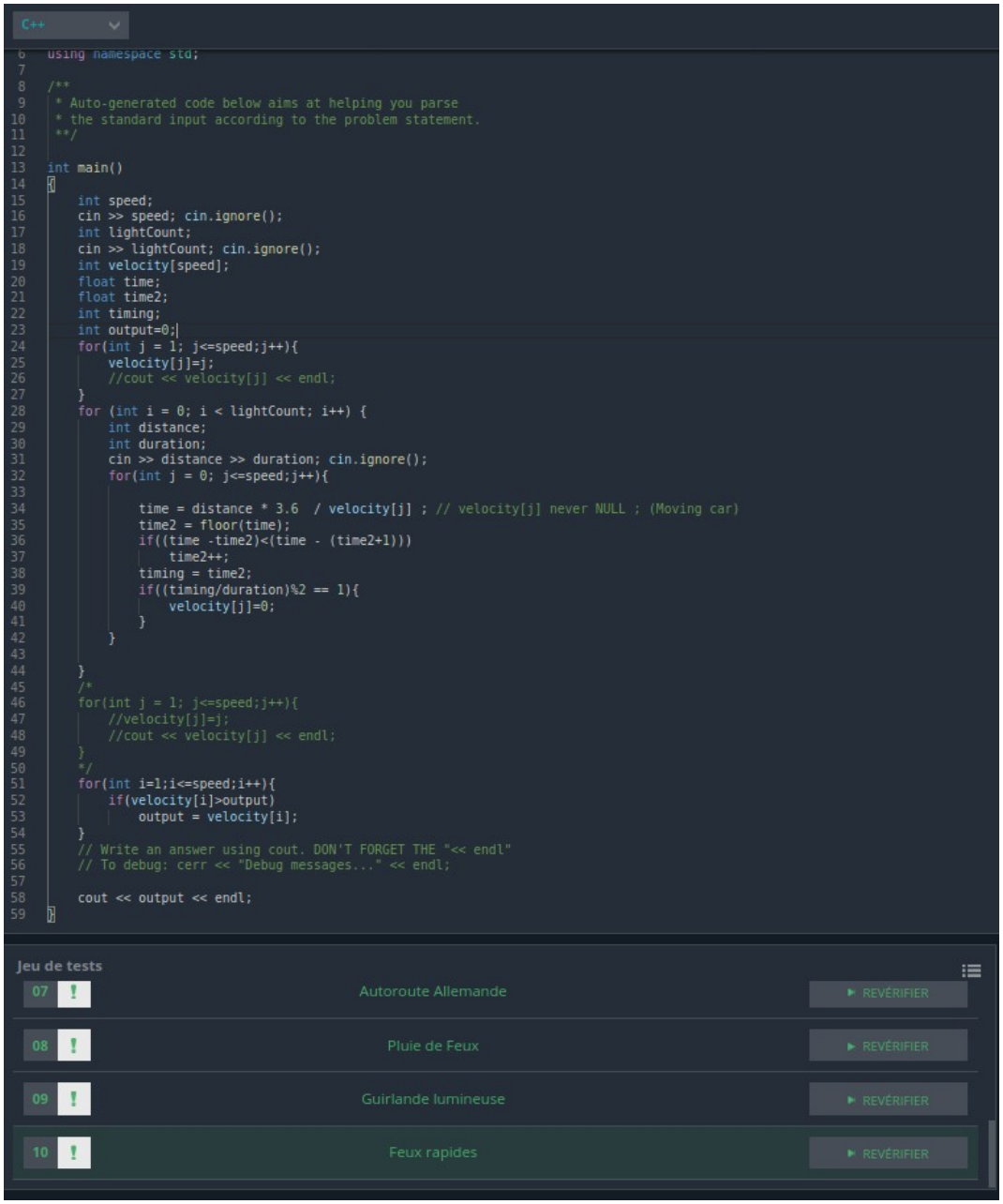
```cpp
int main()
{
    int speed;
    cin >> speed; cin.ignore();
    int lightCount;
    cin >> lightCount; cin.ignore();
    int velocity[speed];
    int time;
    int output=0;
    for(int j = 1; j<=speed;j++){
        velocity[j]=j;
        //cout << velocity[j] << endl;
    }
    for (int i = 0; i < lightCount; i++) {
        int distance;
        int duration;
        cin >> distance >> duration; cin.ignore();
        for(int j = 0; j<=speed;j++){
            time = distance / velocity[j]; // velocity[j] never NULL ; (Moving car)
            if((time/duration)%2 == 1){
                velocity[j]=0;
            }
        }

    }
    for(int j = 1; j<=speed;j++){
        //velocity[j]=j;
       // cout << velocity[j] << endl;
    }
    for(int i=1;i<=speed;i++){
        if(velocity[i]>output)
            output = velocity[i];
    }
    // Write an answer using cout. DON'T FORGET THE "<< endl"
    // To debug: cerr << "Debug messages..." << endl;

    cout << output << endl;
}
```

I have passed the first test but not the second because of the next issues :
- Velecity transformation to m/s.
- Variables transformation to float to do division and multiplication .
- Switch to integer to test.

And it has finally worked :



```cpp
using namespace std;

/**
 * Auto-generated code below aims at helping you parse
 * the standard input according to the problem statement.
 **/

int main()
{
    int speed;
    cin >> speed; cin.ignore();
    int lightCount;
    cin >> lightCount; cin.ignore();
    int velocity[speed];
    float time;
    float time2;
    int timing;
    int output=0;
    for(int j = 1; j<=speed;j++){
        velocity[j]=j;
        //cout << velocity[j] << endl;
    }
    for (int i = 0; i < lightCount; i++) {
        int distance;
        int duration;
        cin >> distance >> duration; cin.ignore();
        for(int j = 0; j<=speed;j++){

            time = distance * 3.6  / velocity[j] ; // velocity[j] never NULL ; (Moving car)
            time2 = floor(time);
            if((time -time2)<(time - (time2+1)))
                time2++;
            timing = time2;
            if((timing/duration)%2 == 1){
                velocity[j]=0;
            }
        }

    }
    /*
    for(int j = 1; j<=speed;j++){
        //velocity[j]=j;
        //cout << velocity[j] << endl;
    }
    */
    for(int i=1;i<=speed;i++){
        if(velocity[i]>output)
            output = velocity[i];
    }
    // Write an answer using cout. DON'T FORGET THE "<< endl"
    // To debug: cerr << "Debug messages..." << endl;

    cout << output << endl;
}
```

Jeu de tests

07 !	Autoroute Allemande	▶ REVÉRIFIER
08 !	Pluie de Feux	▶ REVÉRIFIER
09 !	Guirlande lumineuse	▶ REVÉRIFIER
10 !	Feux rapides	▶ REVÉRIFIER

However, I have decided to add some modifications. (next page)

```cpp
int main()
{
    int speed;
    cin >> speed; cin.ignore();
    if((speed > 200)||(speed < 1) )
        exit(1);
    int lightCount;
    cin >> lightCount; cin.ignore();
    if((lightCount > 9999)||(lightCount < 1) )
    exit(1);
    int * velocity =(int*) malloc(sizeof(int) * speed);
    float time;
    float time2;
    int timing;
    int output=0;
    for(int j = 1; j<=speed;j++){
        velocity[j]=j;
    }
    for (int i = 0; i < lightCount; i++) {
        int distance;
        int duration;
        cin >> distance >> duration; cin.ignore();
        if((distance > 9999)||(distance < 1)||(duration > 9999)||(duration < 1) )
        exit(1);
        for(int j = 0; j<=speed;j++){

            time = distance * 3.6  / velocity[j] ; // velocity[j] never NULL ; (Moving car)
            time2 = floor(time);
            if((time -time2)<(time - (time2+1)))
                time2++;
            timing = time2;
            if((timing/duration)%2 == 1){
                velocity[j]=0;
            }
        }

    }

    for(int i=1;i<=speed;i++){
        if(velocity[i]>output)
            output = velocity[i];
    }

    cout << output << endl;
}
```

I have added extra conditions to only get the expected sizes of our input values. However, some of the used tests do not respect those constraints and as expected the code failed to pass them.

Finally :

- Finding the proper solution took me two hours.
- Coding took one and a half hour.