```cpp
#include<iostream>
#include<iomanip>

#include "question4.hpp"
#include "question6.hpp"

std::ostream& operator<<(std::ostream& os, boost::numeric::ublas::matrix<int> mat) { // Fonction permettant l'impression d'une matrice
    int rows = mat.size1(),
        cols = mat.size2();

    for(int row = 0 ; row < rows ; ++row) {
        for(int col = 0 ; col < cols ; ++col) {
            if(col == 0) {
                if(row == 0) {
                    os << "⎡ ";
                } else if(row == rows - 1) {
                    os << "⎣ ";
                } else {
                    os << "⎢ ";
                }
            }

            os << std::setw(4) << mat(row, col) << " ";

            if(col == cols - 1) {
                if(row == 0) {
                    os << "⎤" << std::endl;
                } else if(row == rows - 1) {
                    os << "⎦" << std::endl;
                } else {
                    os << "⎢" << std::endl;
                }
            }
        }
    }

    return os;
}

template<typename T>
std::ostream& operator<<(std::ostream& os, std::vector<T> vec) {
    int vecSize = vec.size();
    for(int i = 0 ; i < vecSize ; ++i) {
        if(i == 0) {
            os << "⎛";
        } else if(i == vecSize - 1) {
            os << "⎝";
        } else {
            os << "⎜";
        }

        os << vec[i];

        if(i == 0) {
            os << "⎞" << std::endl;
        } else if(i == vecSize - 1) {
            os << "⎠" << std::endl;
        } else {
            os << "⎜" << std::endl;
        }
    }
    return os;
}

int main(int argc, char const* argv[]) {
    std::vector<std::tuple<int, int>> carnet(10);
    carnet[0] = std::tuple<int, int>{20, 25};
    carnet[1] = std::tuple<int, int>{20, 25};
    carnet[2] = std::tuple<int, int>{70, 65};
    carnet[3] = std::tuple<int, int>{10, 15};
    carnet[4] = std::tuple<int, int>{10, 5};
    carnet[5] = std::tuple<int, int>{40, 35};
    carnet[6] = std::tuple<int, int>{10, 15};
    carnet[7] = std::tuple<int, int>{80, 75};
    carnet[8] = std::tuple<int, int>{10, 15};
    carnet[9] = std::tuple<int, int>{40, 15};

    boost::numeric::ublas::matrix<int> dynaMat = question4::allonsY(carnet, 100);
    std::vector<int> spots = question6::computeSpots(dynaMat);

    //std::cout << dynaMat << std::endl
              //<< spots << std::endl << std::endl;

    // Question 7
    std::cout << "Sous-ensemble de spots de gain total maximum :" << std::endl
              << spots << std::endl
              << "Gain total maximum :" << std::endl
```

```
                << dynaMat(dynaMat.size1() - 1, dynaMat.size2() - 1) << std::endl;
    return 0;
}
```