

```

#include<iostream>

using namespace std;

int multMatrix(int *vec){
    int size = 7;
    int matrix[size][size];

    for (int i = 1; i < size; ++i)
        matrix[i][i] = 0;

    for (int h = 2; h < size; ++h){
        for (int i = 1; i < size - h + 1; i++){
            int j = i + h - 1;
            matrix[i][j] = 99999;
            for (int k = i; k < j; k++){
                int temp = matrix[i][k] + matrix[k + 1][j] + vec[i - 1] * vec[k] * vec[j];
                if (temp < matrix[i][j])
                    matrix[i][j] = temp;
            }
        }
    }
    return matrix[1][size - 1];
}

int main(){

    int vec[7] = {30, 35, 15, 5, 10, 20, 25};

    cout << multMatrix(vec);
}

```

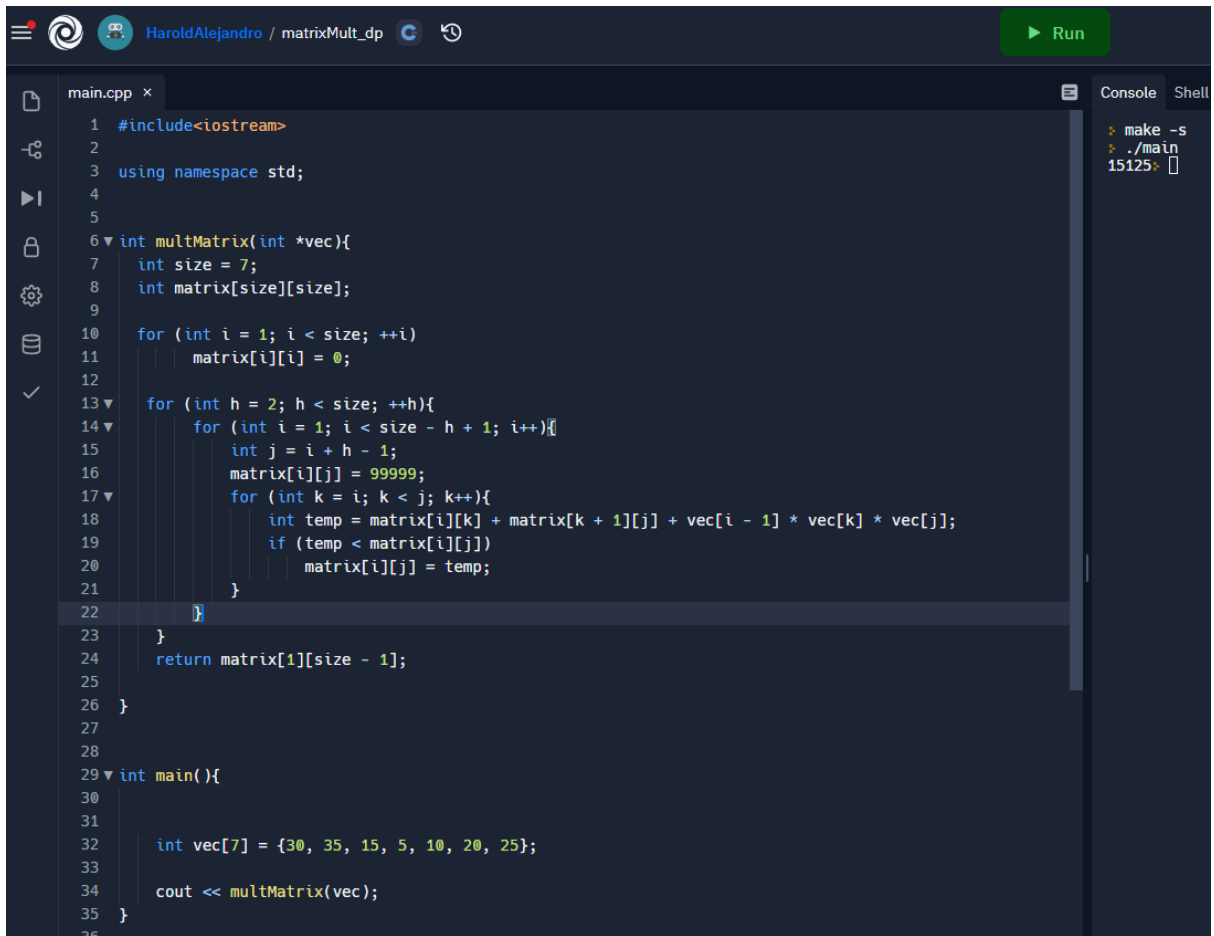
explicación:

pasos para resolver el problema:

1. crear una matriz de nxn
2. llenar la matriz con valores infinitos
3. llenar la diagonal con 0
4. llenar la matriz con los valores de la matriz de la multiplicacion de matrices
5. retornar el valor de la matriz en la posicion [1][size-1]

complejidad temporal:  $O(n^3)$

capturas de pantalla:



The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a `multMatrix` function that takes a vector and returns a matrix. The matrix is initialized with zeros, and then a nested loop calculates its values based on a vector and a previous matrix state. The `main` function initializes a vector and prints the result of `multMatrix`.

```
1  #include<iostream>
2
3  using namespace std;
4
5
6  int multMatrix(int *vec){
7      int size = 7;
8      int matrix[size][size];
9
10     for (int i = 1; i < size; ++i)
11         matrix[i][i] = 0;
12
13     for (int h = 2; h < size; ++h){
14         for (int i = 1; i < size - h + 1; i++){
15             int j = i + h - 1;
16             matrix[i][j] = 99999;
17             for (int k = i; k < j; k++){
18                 int temp = matrix[i][k] + matrix[k + 1][j] + vec[i - 1] * vec[k] * vec[j];
19                 if (temp < matrix[i][j])
20                     matrix[i][j] = temp;
21             }
22         }
23     }
24     return matrix[1][size - 1];
25 }
26
27
28
29 int main(){
30
31
32     int vec[7] = {30, 35, 15, 5, 10, 20, 25};
33
34     cout << multMatrix(vec);
35 }
36
```

The console output shows the execution of the program:

```
make -s
./main
15125
```