

1. Cambio de monedas

Return *the fewest number of coins that you need to make up that amount*. If that amount of money cannot be made up by any combination of the coins, return -1.

```
#include <iostream>
#include <vector>

using namespace std;

int cambiar(vector<int> &coins, int amount)
{
    vector<int> minMonedas(amount + 1);

    //inicializamos la cantidad de monedas
    for (int i = 1; i < minMonedas.size(); i++)
        minMonedas[i] = amount + 1;

    for (int i = 1; i < minMonedas.size(); i++)
    {
        // se busca el valor minimo
        for (auto &&v : coins)
        {
            // verificamos que el valor de moneda sea valido
            if (i - v < 0)
                continue;
            minMonedas[i] = min(minMonedas[i - v] + 1, minMonedas[i]);
        }
    }

    return minMonedas.back() != amount + 1 ? minMonedas.back() : -1;
}

int main()
{
    vector<int> monedas = {1, 2, 5};
    int amount = 9;
    cout << cambiar(monedas, amount) << endl;
    return 0;
}
```

PS C:\xampp\htdocs\algoritmos2\recursivo> g++ .\cambio_moneda.cpp

PS C:\xampp\htdocs\algoritmos2\recursivo> .\a.exe

2. Suma mínima de trayectoria descendente

A falling path starts at any element in the first row and chooses the element in the next row that is either directly below or diagonally left/right. Specifically, the next element from position (row, col) will be (row + 1, col - 1), (row + 1, col), or (row + 1, col + 1).

```
#include <iostream>
#include <vector>

using namespace std;

int sumaMinima(vector<vector<int>> &matrix)
{
    vector<vector<int>> minSumaa;
    minSumaa = matrix;
    int minimo = 0;

    for (int i = 1; i < matrix.size(); i++)
    {
        for (int j = 0; j < matrix[0].size(); j++)
        {
            minimo = minSumaa[i - 1][j];

            if (j > 0)
                minimo = min(minimo, minSumaa[i - 1][j - 1]);

            if (j < matrix[0].size() - 1)
                minimo = min(minimo, minSumaa[i - 1][j + 1]);

            minSumaa[i][j] = matrix[i][j] + minimo;
        }
    }

    minimo = minSumaa.back()[0];
    for (int i = 1; i < minSumaa[0].size(); i++)
    {
        if (minSumaa.back()[i] < minimo)
            minimo = minSumaa.back()[i];
    }
    return minimo;
}
```

```
int main()
{
    vector<vector<int>> matrix;
    matrix.push_back({2, 1, 3});
    matrix.push_back({6, 5, 4});
    matrix.push_back({7, 8, 9});
    cout << sumaMinima(matrix) << endl;
    return 0;
}
```

PS C:\xampp\htdocs\algoritmos2\recursivo> g++ .\suma_minima.cpp

PS C:\xampp\htdocs\algoritmos2\recursivo> .\a.exe

13