Tarea 8

Layla Tame A01192934

// main.cpp

// Tarea 8

// Created by Layla Tame on 3/24/19.

// Copyright © 2019 Layla Tame. All rights reserved.

//Coin Change Program

#include <iostream>

using namespace std;

#define TAM 5

int monedas[TAM] = {1, 2, 5, 10, 20};

void printMatriz(int monedas[TAM], int iSize)

{

for(int i = 0; i<iSize; i++)

{

cout << monedas[i] << " ";

}

cout << endl;

}

int minMonedas1(int iCambio, int monedas[TAM], int minCambio[iCambio])

{

//Inicializar matriz con valores maximos

for(int i = 1; i<=iCambio; i++)

{

minCambio[i] = 99999;

}

//Iterar desde 1 hasta el valor de cambio que se quiere obtener, 0 ya se inicializó

for(int i = 1; i<iCambio+1; i++)

{

//Iterar para todas las monedas en el arreglo

for(int j = 0; j<TAM; j++)

{

//Checar si la moneda es menor al valor que se quiere regresar

if(monedas[j]<=i)

{

//Si el valor existente es menor a la solución actual, guardarlo y sumarle una moneda

if(minCambio[i-monedas[j]] < minCambio[i])

{

minCambio[i] = minCambio[i-monedas[j]] + 1;

}

}

//Ir viendo los cambios en la matriz

//printMatriz(minCambio, iCambio+1);

}

}

return minCambio[iCambio];

}

int minMonedas2(int iCambio, int monedas[TAM], int minCambio[iCambio], int cantMonedas[TAM])

{

//Inicializar matriz con valores maximos

for(int i = 1; i<=iCambio; i++)

{

minCambio[i] = 99999;

}

//Iterar desde 1 hasta el valor de cambio que se quiere obtener, 0 ya se inicializó

for(int i = 1; i<iCambio+1; i++)

{

//Iterar para todas las monedas en el arreglo

for(int j = 0; j<TAM; j++)

{

//Checar si la moneda es menor al valor que se quiere regresar

if(monedas[j]<=i)

{

//Verificar el número de monedas disponibles

if(cantMonedas[j] > 0)

{

//Si el valor existente es menor a la solución actual, guardarlo y sumarle una moneda

if(minCambio[i-monedas[j]] < minCambio[i])

{

minCambio[i] = minCambio[i-monedas[j]] + 1;

cantMonedas[j]--;

}

}

}

//Ir viendo los cambios en la matriz

//printMatriz(minCambio, iCambio+1);

}

}

return minCambio[iCambio];

}

int minMonedas3(int iCambio, int monedas[TAM], int minCambio[iCambio], int cantMonedas[TAM])

{

int camino[iCambio];

//Inicializar matriz con valores maximos

for(int i = 1; i<=iCambio; i++)

{

minCambio[i] = 99999;

camino[i] = 99999;

}

//Iterar desde 1 hasta el valor de cambio que se quiere obtener, 0 ya se inicializó

for(int i = 1; i<iCambio+1; i++)

{

//Iterar para todas las monedas en el arreglo

for(int j = 0; j<TAM; j++)

{

//Checar si la moneda es menor al valor que se quiere regresar

if(monedas[j]<=i)

{

if(cantMonedas[j] > 0)

{

//Si el valor existente es menor a la solución actual, guardarlo y sumarle una moneda

if(minCambio[i-monedas[j]] < minCambio[i])

{

minCambio[i] = minCambio[i-monedas[j]] + 1;

cantMonedas[j]--;

//Almacenar el camino de monedas que se utilizan para llegar a la respuesta

camino[i-1] = j;

}

}

}

//Ir viendo los cambios en la matriz

//printMatriz(minCambio, iCambio+1);

}

}

int numMonedas[TAM];

//Inicializar con 0

for(int i=0; i<TAM; i++)

{

numMonedas[i] = 0;

}

//Obtener la cantidad de monedas de cada tipo utilizadas

//Iterar restando la cantidad de las monedas que se van almacenando

for(int cant = iCambio-1; cant > 0; cant -= monedas[camino[cant]])

{

numMonedas[camino[cant]] += 1;

}

cout << "Numero de monedas: " << endl;

for(int i=0; i<TAM; i++)

{

if(numMonedas[i] > 0)

{

cout << "$" << monedas[i] << ": " << numMonedas[i] << endl;

}

}

return minCambio[iCambio];

}

int main()

{

int iCambio;

//Cantidad infinita de monedas

cout << "-Versión 1-" << endl;

cout << "Cambio a devolver: ";

cin >> iCambio;

//Matriz de número de monedas

int minCambio[iCambio];

//Para cambio de 0, se dan 0 monedas

minCambio[0] = 0;

int respuesta = minMonedas1(iCambio, monedas, minCambio);

if(respuesta != 99999)

{

cout << "Minimo de monedas: " << respuesta << endl;

}

else

{

cout << "No hay monedas para dar el cambio." << endl;

}

//Versión 2 - Cantidad limitada de monedas

cout << endl << "-Versión 2-" << endl;

cout << "Cantidad de monedas para cada tipo: ";

int cantMonedas[TAM];

int iNum;

for(int i = 0; i<TAM; i++)

{

cin >> iNum;

cantMonedas[i] = iNum;

}

respuesta = minMonedas2(iCambio, monedas, minCambio, cantMonedas);

if(respuesta != 99999)

{

cout << "Minimo de monedas: " << respuesta << endl;

}

else

{

cout << "No hay monedas para dar el cambio." << endl;

}

cout << endl << "-Versión 3-" << endl;

respuesta = minMonedas3(iCambio, monedas, minCambio, cantMonedas);

}