#55997721 | JhossepV's solution for [UVA-497]

Status	Length	Lang	Submitted	Open	Share text	RemoteRunId
Accepted	2897	C++11 5.3.0	2024-11-13 14:54:08	☑	☐	29960588

```cpp
#include <iostream>
using namespace std;

int main()
{
  int numeroCasos;
  cin >> numeroCasos;
  cin.ignore();

  string lineaEntrada;
  getline(cin, lineaEntrada);

  // Trabajamos cada caso de prueba de misiles entrantes
  while (numeroCasos--)
  {
    int alturaMisiles[1000];
    int cantidadMisiles = 0;

    // Leemos las alturas de los misiles uno por uno
    // hasta encontrar una línea vacía que separa los casos
    while (getline(cin, lineaEntrada))
    {
      if (lineaEntrada.empty())
        break;
      alturaMisiles[cantidadMisiles++] = stoi(lineaEntrada);
    }

    // Arrays para encontrar la secuencia más larga de misiles que podemos interceptar
    int longitudIntercepciones[1000]; // Guarda cuántos misiles podemos interceptar hasta cada
posición
    int misilAnterior[1000];          // Guarda la posición del misil anterior que interceptamos
    int maximasIntercepciones = 1;    // Cantidad máxima de misiles que podemos interceptar
    int ultimoMisilInterceptado = 0;  // Posición del último misil en la mejor secuencia

    // Inicializra los arrays: cada misil empieza con longitud 1
    for (int i = 0; i < cantidadMisiles; i++)
    {
      longitudIntercepciones[i] = 1;
      misilAnterior[i] = -1;
    }

    // Buscamos la secuencia más larga de misiles que podemos interceptar
    // teniendo en cuenta que cada misil interceptado debe estar a mayor altura que el anterior
    for (int misilActual = 1; misilActual < cantidadMisiles; misilActual++)
    {
      for (int misilPrevio = 0; misilPrevio < misilActual; misilPrevio++)
      {
        if (alturaMisiles[misilActual] > alturaMisiles[misilPrevio] &&
            longitudIntercepciones[misilPrevio] + 1 > longitudIntercepciones[misilActual])
        {
          longitudIntercepciones[misilActual] = longitudIntercepciones[misilPrevio] + 1;
          misilAnterior[misilActual] = misilPrevio;
          if (longitudIntercepciones[misilActual] > maximasIntercepciones)
          {
            maximasIntercepciones = longitudIntercepciones[misilActual];
            ultimoMisilInterceptado = misilActual;
          }
        }
      }
    }

    // Print del total de misiles que podemos interceptar
    cout << "Max hits: " << maximasIntercepciones << endl;

    // Reconstruimos la secuencia de misiles que vamos a interceptar
    int secuenciaIntercepciones[1000];
    int totalIntercepciones = 0;
    for (int i = ultimoMisilInterceptado; i != -1; i = misilAnterior[i])
    {
      secuenciaIntercepciones[totalIntercepciones++] = alturaMisiles[i];
    }

    // Mostramos las alturas de los misiles que interceptaremos, en orden de llegada
    for (int i = totalIntercepciones - 1; i >= 0; i--)
    {
      cout << secuenciaIntercepciones[i] << endl;
    }

    // Agregamos línea en blanco entre casos (excepto después del último caso)
    if (numeroCasos)
      cout << endl;
  }
  return 0;
}
```

Leave a comment