|  |
| --- |
| using System;  using System.Linq;  namespace LadyBugs  {  class Program  {  static void Main()  {  //Reading the size of the field.  int fieldSize = int.Parse(Console.ReadLine());  int[] field = new int[fieldSize];  Console.WriteLine(string.Join(" ", field)); //Remove  //Reading initial indexes separated by " ".  int[] initialIndexes = Console.ReadLine().Split(' ', StringSplitOptions.RemoveEmptyEntries)  .Select(int.Parse)  .ToArray();  Console.WriteLine(string.Join(" ", initialIndexes)); //Remove  for (int i = 0; i < initialIndexes.Length; i++)  {  if (initialIndexes[i] < fieldSize && initialIndexes[i] >= 0)  {  field[initialIndexes[i]] = 1;  }  }  Console.WriteLine(string.Join(" ", field)); //Remove  //Reading commands in format "{ladybug index} {direction}  //{fly length}", until you get the "end" command.  string input = Console.ReadLine();  while (input != "end")  {  string[] command = input.Split(' ', StringSplitOptions.RemoveEmptyEntries);  int bugIndex = int.Parse(command[0]);  if (bugIndex >= 0 && bugIndex < fieldSize)  {  string direction = command[1];  int flyLength = int.Parse(command[2]);  if (field[bugIndex] == 1)  {  field[bugIndex] = 0;  // Checking the direction.  if (direction == "right")  {  while (bugIndex + flyLength < fieldSize && bugIndex + flyLength >= 0)  {  if (field[bugIndex + flyLength] == 0)  {  field[bugIndex + flyLength] = 1;  break;  }  else  {  bugIndex += flyLength;  }  }  }  else if (direction == "left")  {  while (bugIndex - flyLength < fieldSize && bugIndex - flyLength >= 0)  {  if (field[bugIndex - flyLength] == 0)  {  field[bugIndex - flyLength] = 1;  break;  }  else  {  bugIndex -= flyLength;  }  }  }  }  }  //Reading next command.  input = Console.ReadLine();  Console.WriteLine(string.Join(" ", field)); //Remove  }  Console.WriteLine(string.Join(" ", field));  }  }  } |