Ali Baba and the N Houses

Description

Ali Baba has scoped out a line of N houses that you would like to rob tonight. However, he is realized that he wants to avoid robbing two houses that are adjacent to one another because this will minimize the chance that he gets caught. He has also scouted out the hood, so for every house i, he knows that the net worth he will gain from robbing that house is V[i].

Requirements:

Given the number of houses (N) and a list of the net gained value for each consecutive house (V), find the following:

1. **Function#1:** return maximum amount of gain that Ali baba can get.
2. **Function#2:** return indices of the robbed houses (1-based and ordered from left to right). if there're two or more combinations that have the same max total gain, return **ANY of them**

Trace Example: N = 5, V = [5, 2, 1, 3, 1],
Output:

1. Max gained value = 8
2. Robbed Houses = [1, 4]

Function:

First Function:

```
                    int SolveValue(int[] values, int N)
<returns>the maximum amount of money the Ali Baba can get
```

Second Function:

```
               int[] ConstructSolution(int[] values, int N)
<returns>Array of the indices of the robbed houses (1-based and ordered
from left to right)
```

Example

#	Input	Output

1	N = 5, values = [5 2 1 3 1]	`8`
		`[1,4]`

2	N = 8, values = [8 3 5 1 7 6 5 3]	`25`
		`[1,3,5,7]`

4	N = 6, values = [3 2 5 6 6 9]	`18`
		`[1,4,6]`

C# Help

ARRAYS:

Creating 1D array

```
int [] array = new int [size]
```

Creating 2D array

```
int [,] array = new int [size1, size2]
```

Length of 1D array

```
int arrayLength = my1DArray.Length
```

Length of 2D array

```
int array1stDim = my2DArray.GetLength(0)

int array2ndDim = my2DArray.GetLength(1)
```

Sorting single array

Sort the given array in ascending order

```
Array.Sort(items);
```

Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

```
Array.Sort(master, slave);
```