**7 kyu**

**Round To Nearest**

3186% of 2257 of 85[Insight](https://www.codewars.com/users/Insight)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/round-to-nearest/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/round-to-nearest/solutions/csharp)

[Discourse (13)](https://www.codewars.com/kata/round-to-nearest/discuss/csharp)

* Add to Collection
* |
* Share this kata:

You are given a list of numbers (positive / negative integers) and you need to get the nearest entries of a given value.

For example:

The number 10 is given and you need to get the nearest number in this series: 1, 2, 3, 4, 6, 12. The answer should be 12.

C#

int[] intArray = new int[] {1,2,3,4,6,12};

int givenValue = 10;

RoundUp(givenValue, intArray) => {12};

PHP

$intArray = [1,2,3,4,6,12];

$givenValue = 10;

roundUp($givenValue, $intArray) => [12];

If the given number is 5, the result should be {4,6};

C#

int[] intArray = new int[] {1,2,3,4,6,12};

int givenValue = 5;

RoundUp(givenValue, intArray) => {4,6};

PHP

$intArray = [1,2,3,4,6,12];

$givenValue = 5;

roundUp($givenValue, $intArray) => [4, 6];

If the method gets an empty array, it will return an empty array.

C#

int[] intArray = new int[] {};

int givenValue = 42;

RoundUp(givenValue, intArray) => {};

PHP

$intArray = [];

$givenValue = 42;

roundUp($givenValue, $intArray) => [];

<https://www.codewars.com/kata/simple-fun-number-127-prime-operations>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int[] RoundUp(int number, int[] list)

{

List<int> ans = new List<int>();

int min\_dif = int.MaxValue;

int min\_num = 0;

for (int i = 0; i < list.Length; i++)

{

int dif = Math.Abs(number - list[i]);

if (dif < min\_dif)

{

min\_dif = dif;

min\_num = list[i];

ans = new List<int>();

ans.Add(list[i]);

}

else if (dif == min\_dif)

{

ans.Add(list[i]);

}

}

return ans.ToArray();

//return 0;

}

// Driver code

public static void Main()

{

//Assert.AreEqual(new int[] { -80, -160 }, Kata.RoundUp(-120, new int[] { -10, -20, -40, -80, -160, -320 }));

int[] ans = RoundUp(-120, new int[] { -10, -20, -40, -80, -160, -320 });

Console.WriteLine(ans[0] + " " + ans[1]);

Console.ReadLine();

}

}

}