Sort by Absolute Difference

Given an array of **N** elements and a number **K**. The task is to arrange array elements according to the absolute difference with K, i. e., element having minimum difference comes first and so on.
Note : If two or more elements are at equal distance arrange them in same sequence as in the given array.

Example 1:

```
Input: N = 5, K = 7
arr[] = {10, 5, 3, 9, 2}
Output: 5 9 10 3 2
Explanation: Sorting the numbers accoding to
the absolute difference with 7, we have
array elements as 5, 9, 10, 3, 2.
```

Example 2:

```
Input: N = 5, K = 6
arr[] = {1, 2, 3, 4, 5}
Output: 5 4 3 2 1
Explanation: Sorting the numbers according to
the absolute difference with 6, we have array
elements as 5 4 3 2 1.
```

Your Task:
This is a functional problem. You only need to complete the function **sortABS()**. The **printing** is done automatically by the **driver code**.
Expected Time Complexity: O(N log(N)).
Expected Auxiliary Space: O(N).
Constraints:

1 <= N <= 10^5

1 <= K <= 10^5

```
class Solution:

    #Function to sort the array according to difference with given number.
    def sortAbs(self,a, n, k):
        #code here
        # ans = []
        # for i in range(n):
        #     ans.append([a[i],i])
        a.sort(key=lambda x: abs(x-k))
        # return [x[0] for x in ans]
        return a
```