

题目描述

小明今年升学到了小学1年级来到新班级后，发现其他小朋友身高参差不齐，然后就想基于各小朋友和自己的身高差，对他们进行排序，请帮他实现排序。

输入描述

第一行为正整数 h和n，0<h<200 为小明的身高，0<n<50 为新班级其他小朋友个数。  
第二行为n个正整数，h1 ~ hn分别是其他小朋友的身高，取值范围0<hi<200，且n个正整数各不相同。

输出描述

输出排序结果，各正整数以空格分割，  
和小明身高差绝对值最小的小朋友排在前面，  
和小明身高差绝对值最大的小朋友排在后面，  
如果两个小朋友和小明身高差一样，则个子较小的小朋友排在前面。

用例

输入	100 10 95 96 97 98 99 101 102 103 104 105
输出	99 101 98 102 97 103 96 104 95 105
说明	小明身高100，班级学生10个，身高分别为95 96 97 98 99 101 102 103 104 105，按身高差排序后结果为：99 101 98 102 97 103 96 104 95 105。

题目解析

考察多条件排序。

Java算法源码

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3 import java.util.StringJoiner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         int h = sc.nextInt();
10        int n = sc.nextInt();
11
12        Integer[] heights = new Integer[n];
13        for (int i = 0; i < n; i++) heights[i] = sc.nextInt();
14
15        System.out.println(getResult(h, heights));
16    }
17
18    public static String getResult(int h, Integer[] heights) {
19        Arrays.sort(
20            heights,
21            (a, b) -> {
22                int absA = Math.abs(a - h);
23                int absB = Math.abs(b - h);
24
25                if (absA != absB) return absA - absB;
26                else return a - b;
27            });
28
29        StringJoiner sj = new StringJoiner(" ");
30        for (Integer height : heights) {
31            sj.add(height + "");
32        }
33        return sj.toString();
34    }
35 }
```

JS算法源码

```
1  /* JavaScript Node ACH模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5      input: process.stdin,
6      output: process.stdout,
7  });
8
9  const lines = [];
10 rl.on("line", (line) => {
11     lines.push(line);
12 });
13
14 if (lines.length === 2) {
15     const [h, n] = lines[0].split(" ").map(Number);
16     const heights = lines[1].split(" ").map(Number);
17
18     console.log(getResult(h, heights));
19
20     lines.length = 0;
21 }
22
23 function getResult(h, heights) {
24     heights.sort((a, b) => {
25         const absA = Math.abs(a - h);
26         const absB = Math.abs(b - h);
27
28         if (absA != absB) return absA - absB;
29         else return a - b;
30     });
31
32     return heights.join(" ");
33 }
```

Python算法源码

```
1 # 输入获取
2 h, n = map(int, input().split())
3 heights = list(map(int, input().split()))
4
5
6 # 算法入口
7 def getResult():
8     heights.sort(key=lambda x: (abs(x-h), x))
9     return " ".join(map(str, heights))
10
11
12 # 调用算法
13 print(getResult())
```