Lena Sort

HackerRank

Lena developed a sorting algorithm described by the following pseudocode:

```
lena_sort(array nums) {
    if (nums.size <= 1) {
        return nums;
    }
    pivot = nums[0];
    array less;
    array more;
    for (i = 1; i < nums.size; ++i) {
        // Comparison
        if (nums[i] < pivot) {
            less.append(nums[i]);
        }
        else {
            more.append(nums[i]);
        }
    }
    sorted_less = lena_sort(less);
    sorted_more = lena_sort(more);
    ans = sorted_less + pivot + sorted_more;

    return ans;
}
```

We consider a *comparison* to be any time some $nums[i]$ is compared with $pivot$.

You must solve q queries where each query i consists of some len_i and c_i. For each query, construct an array of len_i distinct elements in the inclusive range between 1 and 10^9 that will be sorted by lena_sort in exactly c_i comparisons, then print each respective element of the unsorted array as a single line of len_i space-separated integers; if no such array exists, print `-1` instead.

Input Format

The first line contains a single integer denoting q (the number of queries).
Each line i of the q subsequent lines contains two space-separated integers describing the respective values of len_i (the length of the array) and c_i (the number of comparisons) for query i.

Constraints

- $1 \le q \le 10^5$
- $1 \le len_i \le 10^5$
- $0 \le c_i \le 10^9$
- $1 \le$ the sum of len_i over all queries $\le 10^6$

Output Format

Print the answer to each query on a new line. For each query i, print len_i space-separated integers describing each respective element in an unsorted array that Lena's algorithm will sort in exactly c_i

comparisons; if no such array exists, print `-1` instead.

Sample Input 0

```
2
5 6
5 100
```

Sample Output 0

```
4 2 1 3 5
-1
```

Explanation 0

We perform the following $q = 2$ queries:

1. One array with $len = 5$ elements is $[4, 2, 1, 3, 5]$. The sequence of sorting operations looks like this:

   - Run lena_sort on $[4, 2, 1, 3, 5]$. Compare $pivot = 4$ with 2, 1, 3, and 5 for a total of 4 comparisons. We're then left with $less = [2, 1, 3]$ and $more = [5]$; we only need to continue sorting $less$, as $more$ is sorted with respect to itself because it only contains one element.

   - Run lena_sort on $less = [2, 1, 3]$. Compare $pivot = 2$ with 1 and 3 for a total of 2 comparisons. We're then left with $less = [1]$ and $more = [3]$, so we stop sorting.

   We sorted $[4, 2, 1, 3, 5]$ in $4 + 2 = 6$ comparisons and $c = 6$, so we print `4 2 1 3 5` on a new line.

2. It's not possible to construct an array with $len = 5$ elements that lena_sort will sort in exactly $c = 100$ comparisons, so we print `-1` on a new line.

Sample Input 1

```
3
1 0
4 6
3 2
```

Sample Output 1

```
1
4 3 2 1
2 1 3
```

Explanation 1

We perform the following $q = 3$ queries:

1. We want an array with $len = 1$ element that lena_sort sorts in $c = 0$ comparisons; any array with 1 element is already sorted (i.e., lena_sort performs 0 comparisons), so we choose $[1]$ as our array and print `1` on a new line.

2. One array with $len = 4$ elements is $[4, 3, 2, 1]$; sorting it with lena_sort looks like this:

- lena_sort on $[4, 3, 2, 1]$. Compare $pivot = 4$ with 3, 2, and 1 for a total of 3 comparisons. We're then left with $less = [3, 2, 1]$ and $more = []$; we only need to continue sorting $less$, as $more$ is empty.

- Run lena_sort on $less = [3, 2, 1]$. Compare $pivot = 3$ with 2 and 1 for a total of 2 comparisons. We're then left with $less = [1, 2]$ and $more = []$, so we only continue sorting $less$.

- Run lena_sort on $less = [2, 1]$. Compare $pivot = 2$ with 1 for a total of 1 comparison. We then stop sorting, as $less = [1]$ and $more = []$.

We sorted $[4, 3, 2, 1]$ in $3 + 2 + 1 = 6$ comparisons and $c = 6$, so we print `4 3 2 1` on a new line.

3. One array with $len = 3$ elements is $[2, 1, 3]$. When we run lena_sort on it, we compare $pivot = 2$ with 1 and 3 for a total of 2 comparisons. We're then left with $less = [1]$ and $more = [3]$, so we stop sorting.

   We sorted $[2, 1, 3]$ in 2 comparisons and $c = 2$, so we print `2 1 3` on a new line.