## Programming Part

# Insertion Sort (20 pts)

## Problem Description

Given an array **a** with $N$ numbers $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$ (possibly repeated), run the following insertion sort algorithm on the array, as introduced in Lecture 2 of the class, for a budget of $B$ back-moves or until the array is sorted. The back-move is defined as Line 4 of the INSERT routine.

The budget part is defined as follows. We add a condition to Line 3 of the **while** loop in INSERT that checks whether there is still any back-move budget. If so, Line 4 and Line 5 are executed; if not, the **while** loop will not be entered (but *data* will still be inserted in Line 6). Then, print out the content of the array.

Note that there are many different variant versions of the insertion sort algorithm. You are asked to implement the version below *exactly* to produce the correct answer for this problem.

```
INSERTION-SORT(A)
1  for i = 1 to A.length
2        INSERT(A, i)
3  return A

INSERT(A, m)
1  data = A[m]
2  i = m − 1
3  while i > 0 and A[i] > data
4        A[i + 1] = A[i]
5        i = i − 1
6  A[i + 1] = data
```

## Input

The first line includes two integers $N$ and $B$, representing the size of the array and the budget on the number of back-moves. The second line includes $N$ integers, representing the elements of the array $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$. All numbers are separated by a space.

## Output

- If the array can be sorted within $B$ back-moves, output a line of

  `The array is [the content of the array] after [X] back-moves.`

  where `[the content of the array]` lists the final $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$ after sorting, and `[X]` is the actual number of back-moves taken, which would be a unique number using the version of the insertion sort algorithm above.

- If the array is not sorted after $B$ back-moves, output a line of

  `The array is [the content of the array] after [B] back-moves.`

  where `[the content of the array]` lists the intermediate $\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[N]$ after the $B$ back-moves, and `[B]` is just the number $B$.

## Constraint

- $1 \leq N \leq 2^{10}$
- $1 \leq B \leq 2^{20}$
- $-2^{30} \leq \mathbf{a}[n] \leq 2^{30}$ for $n \in \{1, 2, \ldots, N\}$

## Sample Testcases

**Sample Input 1**

```
3 2
1 3 2
```

**Sample Output 1**

```
The array is 1 2 3 after 1 back-moves.
```

**Sample Input 2**

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

**Sample Output 2**

```
The array is 1 2 4 3 after 2 back-moves.
```

**Sample Input 3**

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

**Sample Output 3**

```
The array is 2 1 2 4 after 2 back-moves.
```

## Hint

- By design, you can pass this homework by simulating the insertion sort algorithm properly. There is no need for other arithmetic calculations or cuts.