**Queue:**

Write an algorithm to solve the below logics

1.**First negative integer in every window of size k**

Given an array **A[]**of size **N** and a positive integer **K**, find the first negative integer for each and every window(contiguous subarray) of size **K**.

**Example 1:**

**Input :**

N = 5

A[] = {-8, 2, 3, -6, 10}

K = 2

**Output :**

-8 0 -6 -6

**Explanation :**

First negative integer for each window of size k

**{-8, 2}** = -8

**{2, 3}** = 0 (does not contain a negative integer)

**{3, -6}** = -6

**{-6, 10}** = -6

2. **First non-repeating character in a stream**

Given an input stream of **A**of **n** characters consisting only of lower case alphabets. The task is to find the first non repeating character, each time a character is inserted to the stream. If there is no such character then append '#' to the answer.

**Example 1:**

**Input:** A = "aabc"

**Output:** "a#bb"

**Explanation:** For every character first non

repeating character is as follow-

"a" - first non-repeating character is 'a'

"aa" - no non-repeating character so '#'

"aab" - first non-repeating character is 'b'

"aabc" - first non-repeating character is 'b'

**Searching:**

**1.Search insert position of K in a sorted array**

Given a sorted array **Arr[]**(0-index based) consisting of **N**distinct integers and an integer **k**, the task is to find the index of k, if its present in the array **Arr[]**. Otherwise, find the index where **k** must be inserted to keep the array sorted.

**Example 1:**

**Input:**

N = 4

Arr = {1, 3, 5, 6}

k = 5

**Output:** 2

**Explaination:** Since 5 is found at index 2

as Arr[2] = 5, the output is 2.

**Example 2:**

**Input:**

N = 4

Arr = {1, 3, 5, 6}

k = 2

**Output:** 1

**Explaination:** Since 2 is not present in

the array but can be inserted at index 1

to make the array sorted.

**2.Majority Element**

Given an array **A** of **N** elements. Find the majority element in the array. A majority element in an array A of size N is an **element that appears more than N/2 times in the array**.  
 **Example 1:**

**Input:**

N = 3

A[] = {1,2,3}

**Output:**

-1

**Explanation:**

Since, each element in

{1,2,3} appears only once so there

is no majority element.

**Example 2:**

**Input:**

N = 5

A[] = {3,1,3,3,2}

**Output:**

3

**Explanation:**

Since, 3 is present more

than N/2 times, so it is

the majority element.

**3.Minimum times A has to be repeated such that B is a substring of it**

Given two strings **A**and**B.**Find minimum number of times A has to be repeated such that B is a Substring of it. If **B** can never be a substring then return **-1**.

**Example 1:**

**Input:**

A = "abcd"

B = "cdabcdab"

**Output:**

3

**Explanation:**

Repeating A three times (“abcdabcdabcd”),

B is a substring of it. B is not a substring

of A when it is repeated less than 3 times.

**Example 2:**

**Input:**

A = "ab"

B = "cab"

**Output :**

-1

**Explanation:**

No matter how many times we repeat A, we can't

get a string such that B is a substring of it.