**CP Club 365Days Challenge**

**Date – 22/11/2022**

**Programming language – only C language**

**Problem Statement**

**Code must be in C language only**

<https://www.hackerrank.com/challenges/ctci-array-left-rotation/problem?isFullScreen=true&h_l=interview&playlist_slugs%5B%5D=interview-preparation-kit&playlist_slugs%5B%5D=arrays>

**Your Code**:

// 0x36Day of 0x365Days challenge

// VEDANT BHARAD

// 22-11-2022

#include <assert.h>

#include <ctype.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\* ltrim(char\*);

char\* rtrim(char\*);

char\*\* split\_string(char\*);

int parse\_int(char\*);

int\* rotLeft(int a\_count, int\* a, int d, int\* result\_count) {

    \*result\_count=a\_count;

    for(int loop = 0; loop < d; loop++)

    {

        int temp =a[0];

        for(int loop2 = 0; loop2 < (a\_count-1); loop2++){

            a[loop2]=a[loop2+1];

        }

        a[a\_count-1]=temp;

    }

    return a;

}

int main()

{

    FILE\* fptr = fopen(getenv("OUTPUT\_PATH"), "w");

    char\*\* first\_multiple\_input = split\_string(rtrim(readline()));

    int n = parse\_int(\*(first\_multiple\_input + 0));

    int d = parse\_int(\*(first\_multiple\_input + 1));

    char\*\* a\_temp = split\_string(rtrim(readline()));

    int\* a = malloc(n \* sizeof(int));

    for (int i = 0; i < n; i++) {

        int a\_item = parse\_int(\*(a\_temp + i));

        \*(a + i) = a\_item;}

    int result\_count;

    int\* result = rotLeft(n, a, d, &result\_count);

    for (int i = 0; i < result\_count; i++) {

        fprintf(fptr, "%d", \*(result + i));

        // printf("%d", \*(result + i));

        if (i != result\_count - 1) {

            fprintf(fptr, " ");

            // printf(" ");}

    }

    fprintf(fptr, "\n");

    fclose(fptr);

    return 0;

}

char\* readline() {

    size\_t alloc\_length = 1024;

    size\_t data\_length = 0;

    char\* data = malloc(alloc\_length);

    while (true) {

        char\* cursor = data + data\_length;

        char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

        if (!line) {

            break;}

        data\_length += strlen(cursor);

        if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') {

            break;}

        alloc\_length <<= 1;

        data = realloc(data, alloc\_length);

        if (!data) {

            data = '\0';

            break;}

    }

    if (data[data\_length - 1] == '\n') {

        data[data\_length - 1] = '\0';

        data = realloc(data, data\_length);

        if (!data) {

            data = '\0';}

    } else {

        data = realloc(data, data\_length + 1);

        if (!data) {

            data = '\0';

        } else {

            data[data\_length] = '\0';

        }

    }

    return data;

}

char\* ltrim(char\* str) {

    if (!str) {

        return '\0';}

    if (!\*str) {

        return str;}

    while (\*str != '\0' && isspace(\*str)) {

        str++;}

    return str;

}

char\* rtrim(char\* str) {

    if (!str) {

        return '\0';

    }

    if (!\*str) {

        return str;

    }

    char\* end = str + strlen(str) - 1;

    while (end >= str && isspace(\*end)) {

        end--;

    }

    \*(end + 1) = '\0';

    return str;

}

char\*\* split\_string(char\* str) {

    char\*\* splits = NULL;

    char\* token = strtok(str, " ");

    int spaces = 0;

    while (token) {

        splits = realloc(splits, sizeof(char\*) \* ++spaces);

        if (!splits) {

            return splits;

        }

        splits[spaces - 1] = token;

        token = strtok(NULL, " ");

    }

    return splits;

}

int parse\_int(char\* str) {

    char\* endptr;

    int value = strtol(str, &endptr, 10);

    if (endptr == str || \*endptr != '\0') {

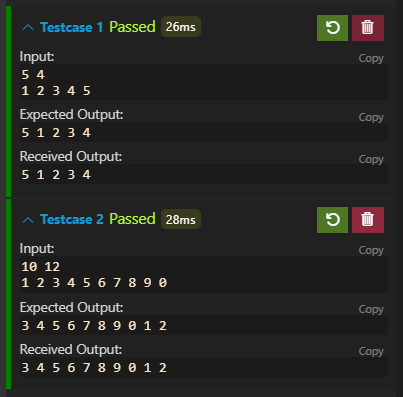
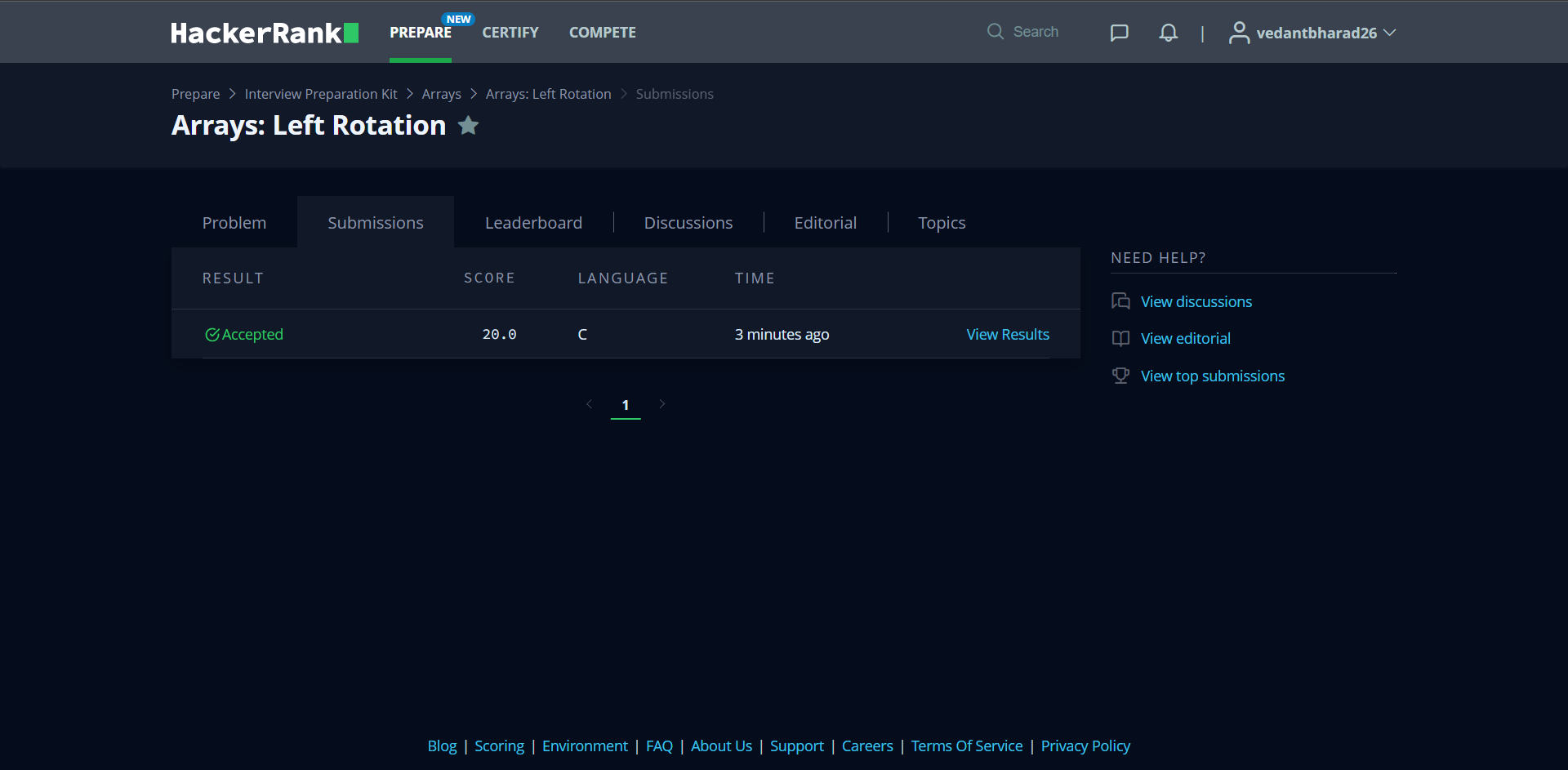
        exit(EXIT\_FAILURE);

    }

    return value;

}

**Output (Screen Shot)**:



**Understanding about problem:**

* In this task there is three inputs

1. Length of array
2. N time Left Rotation
3. Array it self

* In this task I need to do Left Rotation of array and return that.

Note: If you can't understand the problem, feel free to contact us and we'll help you. Please don't copy and paste from anywhere.

ALL THE BEST

Team CP Club