**CP Club 365Days Challenge**

**Date – 23/11/2022**

**Programming language – C**

**Problem Statement**

<https://www.hackerrank.com/challenges/find-digits/problem?isFullScreen=true>

**Your Code**:

// 0x37Day of 0x365Days challenge

// VEDANT BHARAD

// 23-11-2022

#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\*);

int parse\_int(char\*);

int findDigits(int n) {

    int temp=n,con=0;

    while (temp>0)

    {

        if((temp%10)!=0)

        {

            if(n%(temp%10)==0)

            {

                con++;

            }

        }

        temp=temp/10;

    }

    return con;

}

int main()

{

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

    int t = parse\_int(ltrim(rtrim(readline())));

    for (int t\_itr = 0; t\_itr < t; t\_itr++) {

        int n = parse\_int(ltrim(rtrim(readline())));

        int result = findDigits(n);

        // fprintf(fptr, "%d\n", result);

        printf("%d\n", result);

    }

    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;

}

int parse\_int(char\* str) {

    char\* endptr;

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

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

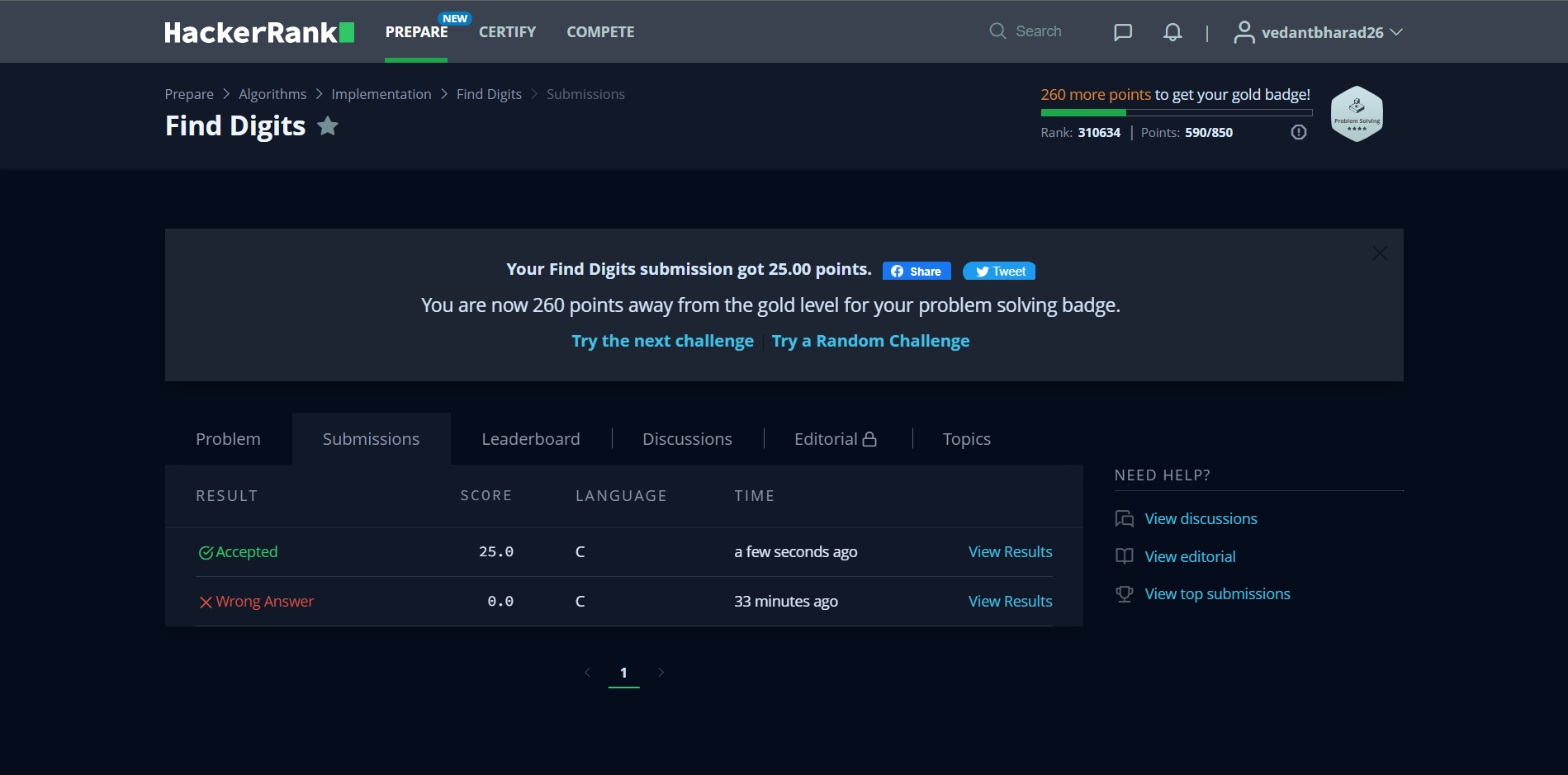
        exit(EXIT\_FAILURE);

    }

    return value;

}

**Output (Screen Shot)**:



**Understanding about problem:**

* In this task there are two inputs

1. Number of test case
2. Number n

* In this task I need to return number which is count of number which are divisor of that n and number which will be checked are every digit of n it self.

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