SRI VASAVI ENGINEERING COLLEGE

**PEDATADEPALLI, TADEPALLIGUDEM.**



Certificate

This is to certify that this is a bonafide record of Practical Work done in **Master in Coding and Competitive Programming-I** by Mr./Miss **SRI SATYA SATTI** bearing Roll No. **21A81A05C3** of **CSE** Branch of **V Semester** during the academic year 2023 to 2024.

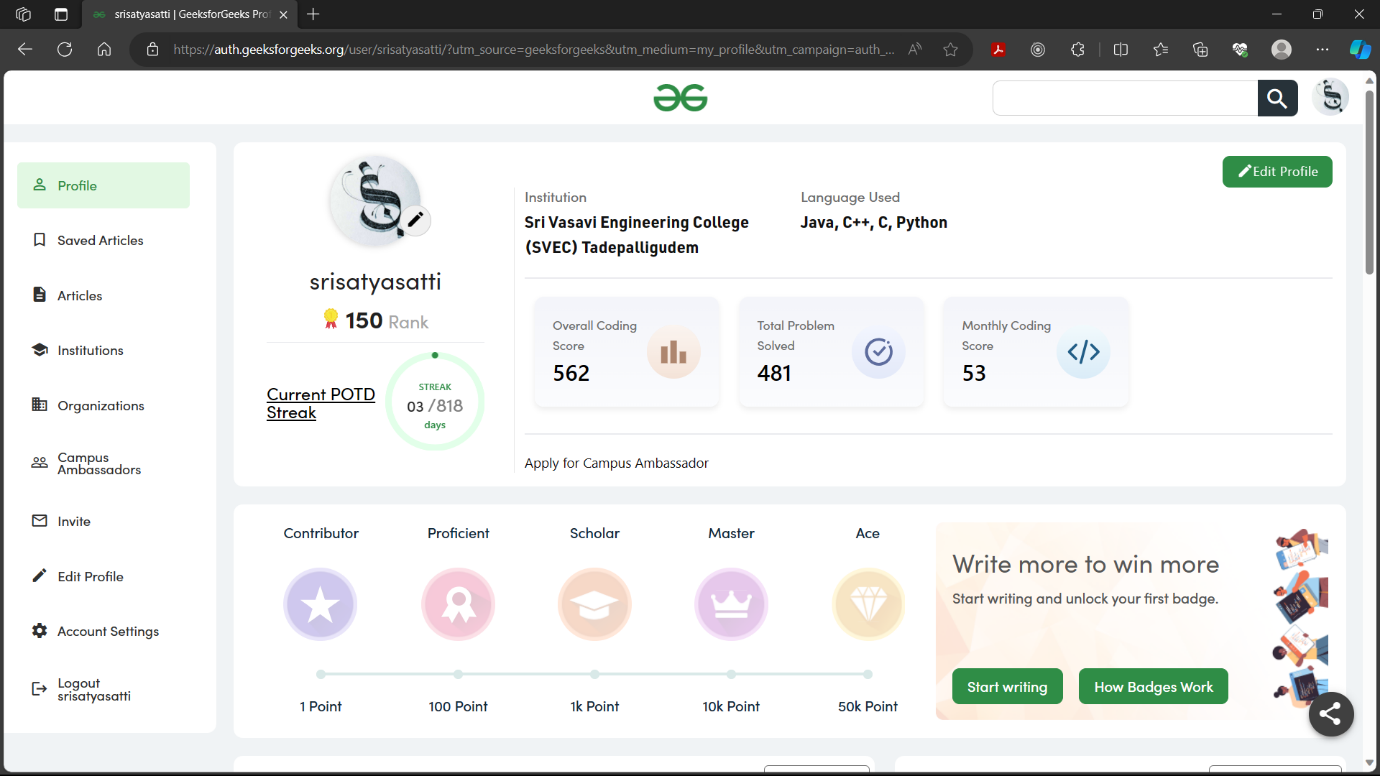
**No. of Experiments Done: 20**

**Faculty In charge of the Laboratory Head of the Department**

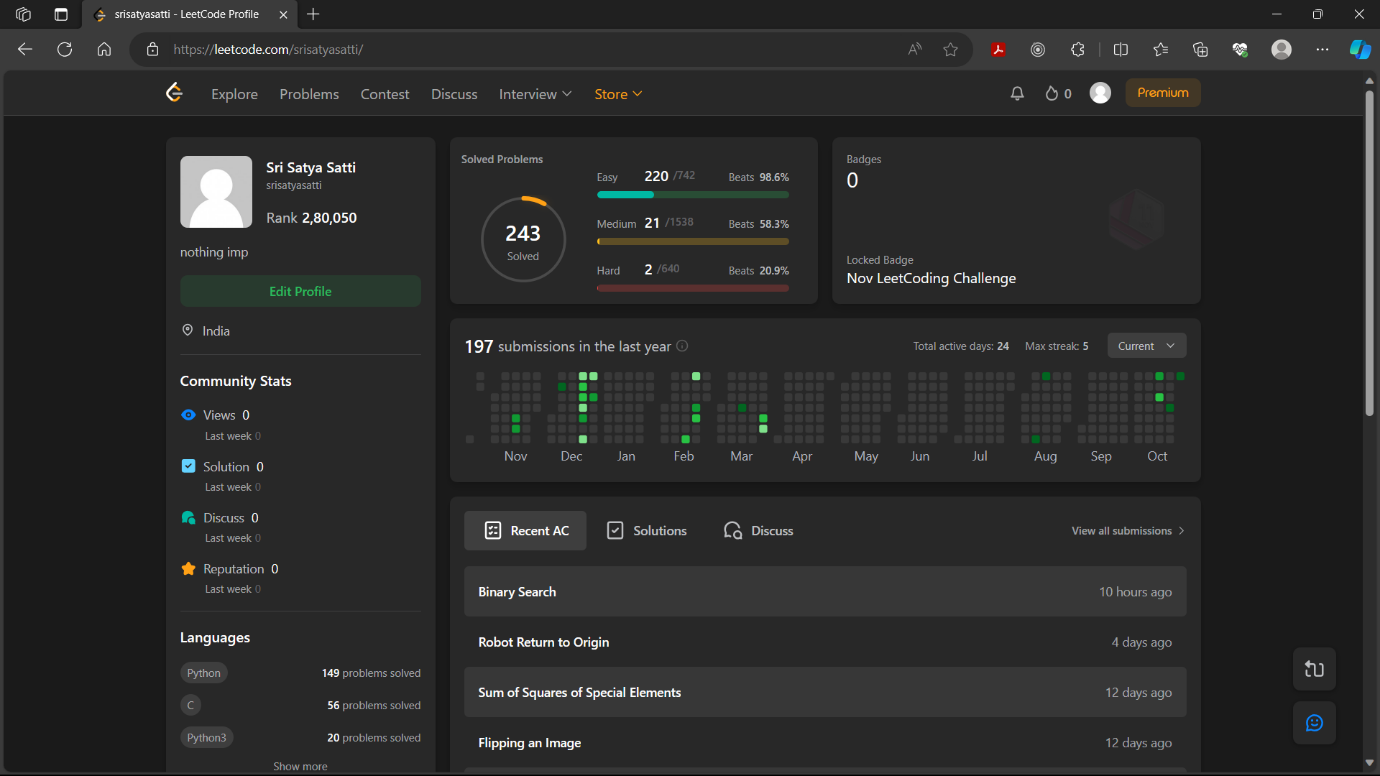
**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment number** | **Name of the experiment** | **Date** | **Signature** |
| Problems on Math Logic | | | |
| 1. | Nth Fibonacci Number |  |  |
| 2. | Sum of all divisors from 1 to n |  |  |
| 3. | Check if the number is balanced |  |  |
| 4. | Trailing zeroes in factorial |  |  |
| 5. | Perfect numbers |  |  |
| programs on Prime numbers | | | |
| 6. | Prime number |  |  |
| 7. | Sum of all prime numbers between 1and n |  |  |
| 8. | Largest prime numbers in a rage |  |  |
| 9. | Damon prime |  |  |
| programs on fibonacci series and series | | | |
| 10. | Last two digit fibonacci |  |  |
| 11. | Modified Fibonacci |  |  |
| 12. | Lucas number |  |  |
| 13. | Padovan sequence |  |  |
| 14. | Geek and his tricky series |  |  |
| programs on arrays | | | |
| 15. | Find duplicates in array |  |  |
| 16. | Peak elements |  |  |
| 17. | Count more than n/k occurrence |  |  |
| 18. | Count odd even |  |  |
| 19. | Count the special |  |  |
| 20. | Frequencies of limited range array elements |  |  |
| 21. | Leaders in array |  |  |
| 22. | Missing number in array |  |  |
| programs on strings | | | |
| 23. | anagram |  |  |
| 24. | Non repeating character |  |  |
| 25. | Find first repeated character |  |  |
| 26. | Reverse word in a given string |  |  |
| 27. | Pangram checking |  |  |
| 28. | Palindrome Sentence |  |  |
| 29. | Check if the number is balanced |  |  |
| 30. | Palindrome String |  |  |
| 31. | Pangram Strings |  |  |
| programs using Hashmap and Hashset | | | |
| 32. | Connell Sequence |  |  |
| 33. | First Repeating Element |  |  |
| 34. | Non-Repeating Element |  |  |
| 35. | Deficient Number |  |  |
| 36. | Geek and his Tricky Series |  |  |
| 37. | First Come First Serve |  |  |
| 38. | First Element to occur k times |  |  |
| 39. | Union of two arrays |  |  |
| 40. | Uncommon characters |  |  |

**Geeks For Geeks Profile:**

****

**Leet Code Profile:**

****

**Problems based on Math Logic:**

### 1.Nth Fibonacci Number

### Given a positive integer n, find the nth fibonacci number. Since the answer can be very large, return the answer modulo 1000000007. Example 1:

### Input: n = 2

### Output: 1

### Example 2:

### Input: n = 5

### Output: 5

### 

### 2. Sum of all divisors from 1 to n

### Given a positive integer ****N****., The task is to find the value of ****Σi from 1 to NF(i)**** where function ****F(i)**** for the number ****i**** is defined as the sum of all divisors of ****i****.

### ****Example 1:****

### **Input: N = 4**

### **Output:** 15

### ****Example 2:****

### **Input:**

### N = 5

### **Output:** 21

### 

### 3.Check if the number is balanced

### Given an integer****N****which has odd number of digits, find whether the given number is a balanced or not.

### An odd digit number is called a balanced number if the sum of all digits to the left of the middle digit and the sum of all digits to the right of the middle digit is equal.

### ****Example 1:1:****

### **Input:** N = 1234006

### **Output:** 1

### ****Example 2:****

### **Input:** N = 12345

### **Output:** 0

### 

### 4.Trailing zeroes in factorial

### For an integer ****N**** find the number of trailing zeroes in ****N!.****

### ****Example 1:****

### **Input:**

### N = 5

### **Output:** 1

### ****Example 2:****

### **Input:** N = 4

### **Output:** 0

### 

### 5.Perfect Numbers

### Given a number ****N,**** check if a number is perfect or not. A number is said to be perfect if sum of all its factors excluding the number itself is equal to the number. Return 1 if the number is Perfect otherwise return 0.

### ****Example 1:****

### **Input:**

### **N =** 6

### **Output:** 1

### ****Example 2:****

### **Input:**

### **N =** 10

### **Output:** 0

### 

### programs on Prime numbers:

### 1.Prime Number

### For a given number ****N**** check if it is prime or not. A prime number is a number which is only**divisible by 1 and itself**.

### ****Example 1:****

### **Input:**

### N = 5

### **Output:**

### 1

### ****Example 2:****

### **Input:** N = 25

### **Output:**0

### 

### 2.Sum of all prime numbers between 1 and N

### Given a positive integer ****N****, find the sum of all prime numbers between ****1**** and ****N****(inclusive).

### ****Example 1:****

### **Input:** N = 5

### **Output:** 10

### ****Example 2:****

### **Input:**

### N = 10

### **Output:**

### 17

### 

### 3.Largest prime factor

### Given a number ****N****, the task is to find the largest prime factor of that number.

### ****Example 1:****

### **Input:**

### N = 5

### **Output:** 5

### ****Example 2:****

### **Input:**

### N = 24

### **Output:** 3

### 

### 4.Find Prime numbers in a range

### Given two integers M and N, generate all primes between M and N including M and N.

### ****Example 1:****

### **Input:**

### M=1, N=10

### **Output:**

### 2 3 5 7

### ****Example 2:****

### **Input:**

### M=2, N=5

### **Output:**

### 2,3,5

### 

### 5.Damon Prime

### Damon is a mathematician, and he calls a integer ****N****Damon Prime if ****N+1**** and ****N-1**** both are prime numbers. He will give you an integer ****N****, you need to find whether the number is Damon Prime or not.   **For example:**  4 is a damon prime, 5 is not a damon prime, 102 is a damon prime, 9 is not a damon prime, etc.  ****Example 1:****

### **Input**:

### N = 4

### **Output:**

### "Yes"

### ****Example 2:****

### **Input:**

### N = 5

### **Output:** "No"

### 

### programs on fibonacci series and series:

### 1.Last two digit Fibonacci

### Given a number N. Find the last two digits of the Nth fibonacci number. ****Note:**** If the last two digits are 02, return 2.

### ****Example 1:****

### **Input:** N = 13

### **Output:** 33

### ****Example 2:****

### **Input:** N = 255

### **Output:** 70

### 

### 2.Modified Fibonacci

### Given 4 integers A, B, C and N, find the value of F(N) such that F(1) = A + B  F(2) = B + C  F(N) = F(N-1) - F(N-2),  for N > 2.

### ****Example 1:****

### **Input:**

### N = 2, A = 2, B = 3, C = 4

### **Output:** 7

### ****Example 2:****

### **Input:**

### N = 3, A = 2, B = 3, C = 4

### **Output:** 2

### 

### 3.Lucas Number

### A Lucas Number is a number which is represented by the following recurrence Ln = Ln-1 + Ln-2 for**n>1** L0 = 2 L1 = 1

### Given a number **N,** find the N**th** lucas number.

### **Note:** Since the output may be very large calculate the answer **modulus 10^9+7.**

### ****Example 1:****

### **Input:** N = 5

### **Output:** 11

### ****Example 2:****

### **Input**: N = 7

### **Output:** 29

### 

### 4.Padovan Sequence

### A **Padovan Sequence** is a sequence which is represented by the following recurrence relation P(N) = P(N-2) + P(N-3) P(0) = P(1) = P(2) = 1 Given a number N, find the Nth number in this sequence.

### **Note:** Since the output may be very large, compute the answer modulus 10^9+7. ****Example 1:****

### **Input:**

### N = 3

### **Output:** 2

### ****Example 2:****

### **Input**: N = 4

### **Output:** 2

### 

### 5.Geek and his Tricky Series

### Given a series with starting 6 members of the series. Given an integer ****n**** find the ****nth**** term of this series modulo 109+7. Series: 7, 15, 32, 67, 138, 281, ............

### ****Example 1:****

### **Input:** n = 2

### **Output:** 15

### 

### programs on arrays ::

### 1.Find duplicates in an array

### Given an array **a** of size **N** which contains elements from **0** to **N-1**, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing **[-1]**.

### **Note:** The extra space is only for the array to be returned. Try and perform all operations within the provided array.

### ****Example 1:****

### **Input:** N = 4

### a[] = {0,3,1,2}

### **Output:** -1

### ****Example 2:****

### **Input:** N = 5

### a[] = {2,3,1,2,3}

### **Output:** 2 3

### 

### 2.Peak element

### An element is called a peak element if its value is not smaller than the value of its adjacent elements(if they exists). Given an array ****arr[]**** of size **N**, **Return the index of any one of its peak elements**. ****Note:****The generated output will always be 1 if the index that you return is correct. Otherwise output will be 0.

### ****Example 1:****

### **Input:** N = 3

### arr[] = {1,2,3}

### **Possible Answer:** 2

### **Generated Output:** 1

### ****Example 2:****

### **Input:** N = 3

### arr[] = {3,4,2}

### **Possible Answer:** 1

### **Output:** 1

### 

### 3.Count More than n/k Occurences

### Given an array ****arr****[] of size ****N**** and an element ****k****. The task is to find all elements in array that appear more than ****n/k**** times.

### ****Example 1:****

### **Input:** N = 8

### arr[] = {3,1,2,2,1,2,3,3}

### k = 4

### **Output:** 2

### ****Example 2:****

### **Input:**

### N = 4

### arr[] = {2,3,3,2}

### k = 3

### **Output:** 2

### 

### 4.Count Odd Even

### Given an array ****A[]****of ****N****elements. The task is to count number of ****even****and ****odd****elements in the array.

### ****Example:****

### **Input:** N = 5

### A[] = 1 2 3 4 5

### **Output:** 3 2

### 

### 5.Count the Specials

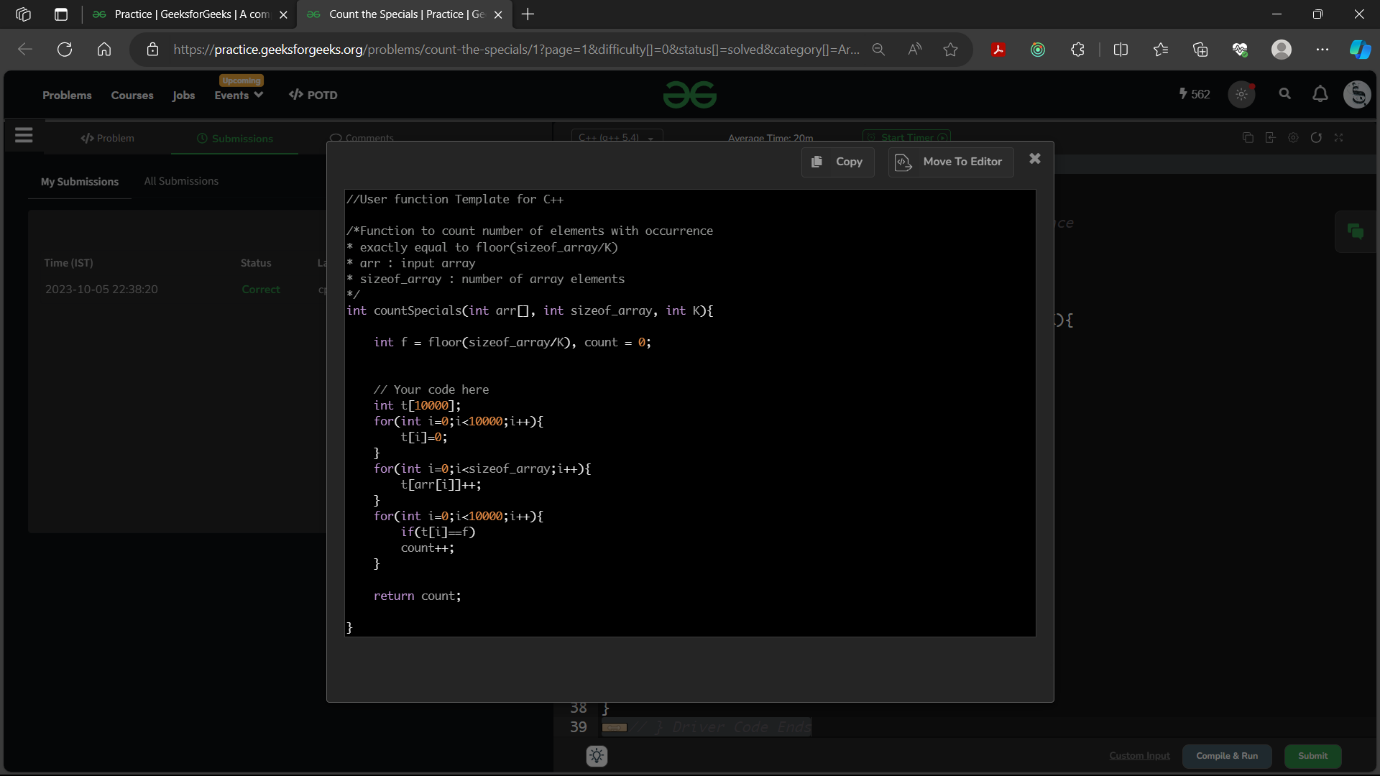
### Given an array ****A**** (may contain duplicates) of ****N**** elements and a positive integer ****K****. The task is to count the number of elements that occurs exactly ****floor(N/K)**** times in the array.

### ****Example:****

### **Input:** N = 5 K = 2

### A[] = 1 4 1 2 4

### **Output:**2



### 6.Frequencies of Limited Range Array Elements

### Given an array ****arr[]**** of ****N**** positive integers which can contain integers from ****1 to P**** where elements can be repeated or can be absent from the array. Your task is to count the frequency of all numbers from ****1 to N****. Make in-place changes in ****arr[],**** such that **arr[i] = frequency(i).** Assume 1-based indexing. ****Note:**** The elements greater than N in the array can be ignored for counting and **do modify the array** **in-place.**

### ****Example 1:****

### **Input:** N = 5

### arr[] = {2, 3, 2, 3, 5}

### P = 5

### **Output:**

### 0 2 2 0 1

### ****Example 2:****

### **Input:** N = 4

### arr[] = {3,3,3,3}

### P = 3

### **Output:** 0 0 4 0

### 

### 7.Leaders in an array

### Given an array A of positive integers. Your task is to find the leaders in the array. An element of array is leader if it is greater than or equal to all the elements to its right side. The rightmost element is always a leader.

### ****Example 1:****

### **Input:** n = 6

### A[] = {16,17,4,3,5,2}

### **Output:** 17 5 2

### ****Example 2:****

### **Input:** n = 5

### A[] = {1,2,3,4,0}

### **Output:** 4 0

### 

### 8.Missing number in array

### Given an array of size ****N-1**** such that it only contains distinct integers in the range of ****1 to N****. Find the missing element.

### ****Example 1:****

### **Input:**

### N = 5

### A[] = {1,2,3,5}

### **Output:** 4

### ****Example 2:****

### **Input:**

### N = 10

### A[] = {6,1,2,8,3,4,7,10,5}

### **Output:** 9

### 

### programs on strings::

### 1.Anagram

### Given two strings **a**and **b**consisting of lowercase characters. The task is to check whether two given strings are an anagram of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example, act and tac are an anagram of each other.

### If the strings are anagrams you have to **return True or else return False**

### **|s|**represents the length of string s.

### ****Example 1:****

### ****Input:**** a = geeksforgeeks, b = forgeeksgeeks

### ****Output:**** YES

### ****Example 2:****

### ****Input:**** a = allergy, b = allergic

### ****Output:**** NO

### 

### 2.Non Repeating Character

### Given a string ****S**** consisting of ****lowercase****Latin Letters. Return the first non-repeating character in S. If there is no non-repeating character, return ****'$'.****

### ****Example 1:****

### **Input:**

### S = hello

### **Output:** h

### ****Example 2:****

### **Input:**

### S = zxvczbtxyzvy

### **Output:** c

### 

### 3.Find first repeated character

### Given a string ****S****. The task is to find the first repeated character in it. We need to find the character that occurs more than once and whose index of second occurrence is smallest. S contains only lowercase letters.

### ****Example 1:****

### **Input:** S="geeksforgeeks"

### **Output:** e

### ****Example 2:****

### **Input:**

### S="hellogeeks"

### **Output:** l

### 

### 4.Reverse words in a given string

### Given a String S, reverse the string without reversing its individual words. Words are separated by dots.

### ****Example 1:****

### **Input:**

### S = i.like.this.program.very.much

### **Output:** much.very.program.this.like.i

### ****Example 2:****

### **Input:**

### S = pqr.mno

### **Output:** mno.pqr

### 

### 5.Panagram Checking

### Given a string check if it is Panagram or not. A panagram is a sentence containing every letter in the English Alphabet (either lowercase or uppercase or both). For example, we say the letter **A is present**in the string if either '****a****' is present or '****A****' is present.

### ****Example 1:****

### **Input:**

### S = Bawds jog, flick quartz, vex nymph

### **Output:** 1

### ****Example 2:****

### **Input:** S = sdfs

### **Output:** 0

### 

### 6.Palindrome Sentence

### Given a single sentence ****s****, check if it is a palindrome or not. Ignore white spaces and any other character you may encounter.

### ****Example 1:****

### **Input:**

### s = race car.

### **Output:** 1

### ****Example 2:****

### **Input:**

### s = hello world.

### **Output:** 0

### 

### 7.Check if the number is balanced

### Given an integer****N****which has odd number of digits, find whether the given number is a balanced or not.

### An odd digit number is called a balanced number if the sum of all digits to the left of the middle digit and the sum of all digits to the right of the middle digit is equal.

### ****Example 1:****

### **Input:**

### N = 1234006

### **Output:** 1

### ****Example 2:****

### **Input:**

### N = 12345

### **Output:** 0

### 

### 7.Palindrome String

### Given a string ****S****, check if it is palindrome or not.

### ****Example 1:****

### **Input:**

### S = "abba"

### **Output:** 1

### ****Example 2:****

### **Input:**

### S = "abc"

### **Output:** 0

### 

### 8.Pangram Strings

### Check if the given string ****S****is a **Panagram**or not. A pangram is a sentence containing every letter in the English Alphabet.

### ****Example 1:****

### **Input:**

### S = "Pack mY box witH fIve dozen liquor jugs"

### **Output:** 1

### ****Example 2:****

### **Input:**

### S = "geeksFORgeeks"

### **Output:** 0

### 

### programs using hashmap and hashset::

### 1.Connell Sequence

### **Connell Sequence** is the sequence formed with the first odd number, i.e 1 as its first term. The subsequent terms of the sequence are made up of the first two even numbers, i.e 2 and 4, followed by the next three odd numbers, i.e 5, 7 and 9, followed by the next four even numbers, i.e 10, 12, 14 and 16 and so on ….  Given an integer n, generate the first n terms of the Connell Sequence.

### ****Example 1:****

### **Input:** n = 6

### **Output**:**** 1 2 4 5 7

### **Example 2:**

### **Input:** n = 2

### **Output:** 1 2

### 

### 

### 2. First Repeating Element

### Given an array arr[] of size n, find the first repeating element. The element should occur more than once and the index of its first occurrence should be the smallest.

### **Note:-**The position you return should be according to 1-based indexing.

### ****Example 1:****

### **Input:** n = 7

### arr[] = {1, 5, 3, 4, 3, 5, 6}

### **Output:** 2 ****Example 2:****

### **Input:**

### n = 4

### arr[] = {1, 2, 3, 4}

### **Output:** -1

### 

### 3.Non-Repeating Element

### Find the first non-repeating element in a given array****arr**** of ****N**** integers. ****Note:**** Array consists of only positive and negative integers and **not zero**.

### ****Example 1:****

### **Input :** arr[] = {-1, 2, -1, 3, 2}

### **Output :** 3

### ****Example 2:****

### **Input :** arr[] = {1, 1, 1}

### **Output :** 0

### 

### 4.Deficient Number

### Given a number x, your task is to find if this number is Deficient number or not. A number x is said to be Deficient Number if sum of all the divisors of the number denoted by divisorsSum(x) is less than twice the value of the number x. And the difference between these two values is called the **deficiency**.

### Mathematically, if below condition holds the number is said to be Deficient: **divisorsSum(x)**< 2\*x **deficiency** = (2\*x) - divisorsSum(x) ****Example 1:****

### **Input**: x = 21

### **Output:** YES

### ****Example 2:****

### **Input:x** = 12

### **Output:**NO

### 

### 5.Geek and his Tricky Series

### Given a series with starting 6 members of the series. Given an integer ****n**** find the ****nth**** term of this series modulo 109+7. Series: 7, 15, 32, 67, 138, 281, ............

### ****Example 1:****

### **Input:** n = 2, **Output:** 15

### 

### 6.First Come First Serve

### CodeMart is a shopping platform and it is distributing gift vouchers to its esteemed users. The voucher can be redeemed by providing a fixed amount of shopping credits to the platform. One credit is sent by a user to the platform by doing one occurance in CodeMart. Since there is a huge rush of people you are required to manage the users on the basis of first come first serve. The user which came first and has **exactly k** occurrences at last is given the voucher first. You are given an array ****arr[ ]****with the id's of ****N**** users in chronological order of their occurances . You are required to print the id of the user which will be given the voucher first. If no such user meets the condition print "-1".

### ****Example 1:****

### **Input:** N = 7 , K = 2

### arr[] = {1, 7, 4, 3, 4, 8, 7}

### **Output:** 7

### ****Example 2:****

### **Input:** N = 6 ,K = 1

### arr[] = {4, 1, 6, 1, 6, 4}

### **Output:** -1

### 

### 7.First element to occur k times

### Given an array of ****N**** integers. Find the first element that occurs at least ****K**** number of times.

### ****Example 1:****

### **Input :**

### N = 7, K = 2

### A[] = {1, 7, 4, 3, 4, 8, 7}

### **Output :** 4

### 

### 8.Union of two arrays

### Given two arrays **a[]** and **b[]** of size **n** and **m** respectively. The task is to find the number of elements in the union between these two arrays.

### Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of element should be printed in the union.

### **Note :**Elements are not necessarily distinct.

### ****Example 1:****

### **Input:**

### 5 3

### 1 2 3 4 5

### 1 2 3

### **Output:** 5

### ****Example 2:****

### **Input:**

### 6 2

### 85 25 1 32 54 6

### 85 2

### **Output:**

### 7

### 

### 9.Uncommon characters

### Given two strings A and B consisting of lowercase english characters. Find the characters that are not common in the two strings.

### **Note** :- Return the string in sorted order.

### ****Example 1:****

### **Input:**

### A = geeksforgeeks

### B = geeksquiz

### **Output:** fioqruz

### ****Example 2:****

### **Input:**

### A = characters

### B = alphabets

**Output:** bclpr

