

Labtest [CSE122](Afternoon Session):

1. Write the following three functions:

(i) **int isPrime(int n)** : Takes an integer as parameter, and return 1 if it is prime, otherwise return 0.

(ii) **int reverseNumber(int n)** : Takes an integer n and returns the number formed by reversing the sequence of digits. For example, if the value of n is 1234, your function should return 4321.

(iii) **int toBase10(char c[],int length, int radix)** : Takes a character array c and the length of the array. This character array will contain a valid number in any base between 2-10. The base of the number will be passed as the parameter radix. The function returns the decimal value of the number. See sample I/O for further clarification.

Sample input for toBase10	Sample Return value
[1,0,1,0], 4, 2	10
[1,2,5], 2, 8	85

Write a program that takes a character array which represents a number in any base between 2-10. Your program should first convert the character array to corresponding decimal value and then print "**Prime**" if the decimal value is prime and should print "**Emirp**" if the reverse of the value is a prime. If neither the number nor its reverse is a prime print "**None**".

<u>Sample Input from Main Function:</u> Enter length of Array: 4 Enter Array Elements : 1 0 1 0 Enter Base: 2	<u>Output:</u> Decimal Value : 10 Status: None
<u>Sample Input from Main Function:</u> Enter length of Array: 3 Enter Array Elements : 141 Enter Base: 8	<u>Output:</u> Decimal Value: 97 Status: Both Prime and Emirp

2. A geometric series is a sequence of numbers where each term after the first is found by multiplying the previous term by a fixed, non-zero number called the common ratio. The formula to calculate the sum of the first n terms of a geometric series is given by:

$$S_n = a \times \frac{1-r^n}{1-r} \text{ for } n > 1 \text{ and } S_n = a \text{ for } n=1$$

Here, a is the first term of the series and r is the common ratio.

Write a program to calculate the sum of a geometric series up to the nth term using global variables and recursion.

3. Prime numbers are numbers that have no other factors besides one and the number itself. Some examples of prime numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 & 37. Emirps are special cases of prime numbers. A prime number is an emirp if it is still prime when its digits are reversed. For example, 37 is an emirp because not only is it prime, but so too is 73 (the reverse of 37). Write a program that will ask for a number. Expect the user to enter only positive integers from 0 to 32,767. Then tell it is: "not prime", "a prime", or "an emirp". The program should continue until a zero is entered.

Sample Input	Corresponding Output
37	37 is an emirp
39	39 is not prime
359	359 is an emirp
32713	32713 is an emirp
773	773 is a prime

4. Write a C program that implements string operations such as compare, concatenate, and string length without using built-in functions. Your program should prompt the user to input two strings: the main string and the substring. Then, it should determine if the substring occurs in the main string. **You cannot use built-in functions other than strlen().**

Sample Input	Sample Output
Mainstring: appleisagoodfruitappleisgood Substring: apple	Found
Mainstring: Todayissaturday Substring: monday	Not Found