upGrad

TechM Full-Stack Software Development

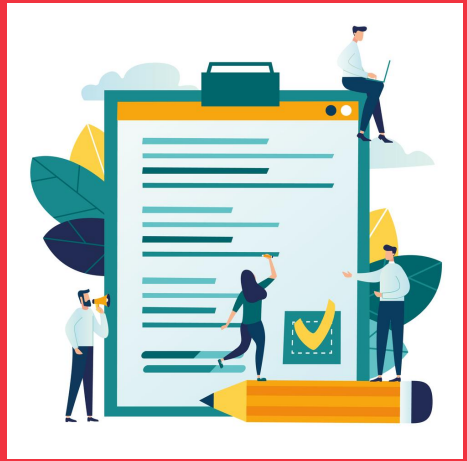
upGrad

Lecture On: Mathematics

Instructor: Arkoprovo Dey

In Last Class, we covered….

- Bit Manipulation



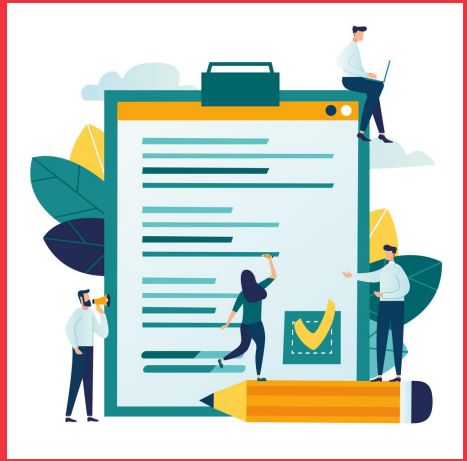
Homework Discussion

1. Write a function which takes a decimal number as an input and prints its octal equivalent.

2. Write a function which takes a octal number as an input and prints its decimal equivalent.

Today's Agenda

1 Build a scientific calculator



Let us build a calculator!

And now let us try to put in a few more fancy mathematical operations into our calculator!

Poll 1 (15 sec.)

2 is a prime number?

1. Yes
2. No

Poll 1 (Answer)

2 is a prime number?

1. **Yes**
2. No

Problem 1: Prime Number check!

Problem: Given a number ‘n’, we want to check if the number is a Prime number of not.

A Prime number is a number that is not divisible by any number apart from 1 and itself!

Eg:

7 is a prime number

24 is NOT a prime number

Poll 2 (15 sec.)

What is the worst case time complexity of finding out is a number is prime or not?

1. O(n^2)
2. O(n)
3. O(sq. root of n)
4. O(1)

Poll 2 (Answer)

What is the worst case time complexity of finding out is a number is prime or not?

1. O(n^2)
2. O(n)
3. **O(sq. root of n)**
4. O(1)

Poll 3 (15 sec.)

What are the factors of 24?

1. 1, 2, 3, 4, 6, 8, 12
2. 1, 2, 3, 12, 24
3. 1, 2, 3, 4, 6, 8, 12, 24
4. 2, 3, 4, 6, 8, 12, 24

Poll 3 (15 sec.)

What are the factors of 24?

1. 1, 2, 3, 4, 6, 8, 12
2. 1, 2, 3, 12, 24
3. **1, 2, 3, 4, 6, 8, 12, 24**
4. 2, 3, 4, 6, 8, 12, 24

Problem 2: Prime factorize a number

Problem: Given a number ‘n’, we print out the prime factorization of the number.

Prime factorization is essentially printing out all the factors of a number apart from 1 and the number itself(that are ‘primes’), such that they multiply to be the number itself.

Eg:

24 can be broken down as 2 x 2 x 2 x 3. Thus the prime factorization of 24 would be “2, 2, 2 and 3”

Poll 4 (15 sec.)

How many numbers will we print out if we try to prime factorize a prime number ‘n’?

1. 1
2. 0
3. ‘n’
4. Sq. root of ‘n’

Poll 4 (Answer)

How many numbers will we print out if we try to prime factorize a prime number ‘n’?

1. 1
2. **0**
3. ‘n’
4. Sq. root of ‘n’

Poll 5 (15 sec.)

What would be the sum of digits for a given number 11234?

1. 12
2. 11
3. 9
4. 10

Poll 5 (Answer)

What would be the sum of digits for a given number 11234?

1. 12
2. **11**
3. 9
4. 10

Problem 3: Find out the sum of digits, max digit, and number of digits or a number

Problem: Given a number ‘n’, we want to figure out the sum of its’ digits, the number of digits in the number, and the maximum digit in the number.

Eg:

2532 - Sum of Digits: 2 + 5 + 3 + 2 = 12, No. of digits: 4, Maximum digit: 5

87239 - Sum of Digits: 8 + 7 + 2 + 3 + 9 = 29, No. of digits: 5, Maximum digit: 9

Poll 6 (15 sec.)

13 is an armstrong number?

1. Yes
2. No

Poll 6 (Answer)

13 is an armstrong number?

1. Yes
2. **No**

Problem 4: Check if a number is an Armstrong number

Problem: Given a number ‘n’, we want to check if the number is an Armstrong number of not.

An Armstrong number is one where the number is equal to the cube of the digits of the number!

Eg:

121 -> 1^3 + 2^3 + 1^3 = 10, which is not equal to 121. So, 121 is NOT an Armstrong number.

407 -> 4^3 + 0^ 3 + 7^3 = 64 + 343 = 407. So, 407 is an Armstrong number.

Poll 7 (15 sec.)

2^3 is equal to?

1. 6
2. 2
3. 3
4. 8

Poll 7 (Answer)

2^3 is equal to?

1. 6
2. 2
3. 3
4. **8**

Problem 5: Raise a number to the power of another number

Problem: Given a number ‘n’ and a number ‘p’, we want to print out n^p.

Eg:

n = 4, p = 3 -> n ^ p = 4 ^ 3 = 64

n = 12, p = 0 -> n ^ p = 12 ^ 0 = 1

Poll 8 (15 sec.)

11 mod 2 is equal to?

1. 0
2. 1
3. 5
4. None of the above

Poll 8 (Answer)

11 mod 2 is equal to?

1. 0
2. **1**
3. 5
4. None of the above

Problem 6: Implement the modulus operator

Problem: Given a number ‘n’ and a number ‘d’, we want to print out the result of n % d.

The ‘%’ operator gives us the remainder after ‘n’ is divided by ‘d’.

Eg:

n = 20, d = 6 -> 20 % 6 = 2

n = 75, d = 5 -> 75 % 5 = 0

n = 37, d = 6 -> 37 % 6 = 1

Problem 7: Check if a number is an Automorphic number

Problem: Given a number ‘n’, we want to check if the number is an Automorphic number of not.

An Automorphic number is one where the square of the number ‘ends’ with the number itself.

Eg:

37 -> 37 ^ 2 = 1369, which does NOT end with 37. So, 37 is NOT an Automorphic number.

76 -> 76 ^ 2 = 57**76**, which ends with ‘76’. So, 76 is an Automorphic number.

25 -> 25 ^ 2 = 6**25**, which ends with ‘25’. So, 25 is an Automorphic number.

Poll 9 (15 sec.)

100 is automorphic number?

1. True
2. False

Poll 9 (Answer)

100 is automorphic number?

1. **True**
2. False

Poll 10 (15 sec.)

Between 1 to 10(both inclusive) how many perfect squares exist?

1. 2
2. 3
3. 1
4. 0

Poll 10 (Answer)

Between 1 to 10(both inclusive) how many perfect squares exist?

1. 2
2. **3**
3. 1
4. 0

Problem 8: Count the number of perfect squares in a range

Problem: Given 2 numbers ‘l’ and ‘u’, we want print all the ‘perfect squares’ that fall in this range.

Perfect squares are numbers that are formed by the square of a number.

Eg:

l = 10, u = 30 -> The perfect squares in this range are - 16 (4 * 4) and 25 (5 * 5) -> Ans: 2

l = 8, u = 17 -> The perfect squares in this range are - 9 (3 * 3) and 16 (4 * 4) -> Ans: 2

PS. Take care of the time complexity while solving this problem!

Homework

1. Write a program to reverse a input number.

2. Write a simple program which will take two numbers as input and prints the GCD and LCM of those two numbers.

3. Write a program which takes a number "n" as an input and prints first n fibonacci numbers.

upGrad

Tasks to complete after the session

Homework Questions
MCQs
Coding Questions

In the next class...

- Recursion

upGrad

#RahoAmbitious

Thank You!