Q1.

**Problem Statement**

You are organizing a gift distribution event, and you want to ensure that each participant receives a unique combination of gifts. You have a set of **x**different types of gifts available, numbered from 1 to x. Each participant should receive a total of **n** gifts. However, you want to distribute the gifts in such a way that each participant receives a combination of gifts that can be expressed as a sum of nth powers of unique natural numbers.

Write a program that takes the total number of available gift types **x** and the power **n**as input. The program should calculate and return the number of unique combinations of gifts that can be distributed following the given criteria using recursion.

**Example 1**

**Input:**

10

2

**Output:**

1

**Explanation:**

x = 10

n = 2

10 = 12 + 32, hence we have only 1 possibility.

**Example 2**

**Input:**

100

2

**Output:**

3

**Explanation:**

x = 100

n = 2

100 = 102 **OR** 62 + 82 **OR** 12 + 32 + 42 + 52 + 72, hence total 3 possibilities.

**Function Specifications:**

int getAllWays(int remainingSum, int power, int base);

**Note:** This is a sample question asked in a HCL interview.

**Input Format**

The first line of input consists of the value of integer x.

The second line of input consists of the value of integer n.

**Output Format**

The output prints the number of ways to express x as a sum of nth powers of unique natural numbers.

**Constraints**

0 < x <=70000

0 < n <=5

**Sample InputSample Output**

10

2

1

**Sample InputSample Output**

100

2

3

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2.

**Problem Statement**

You are exploring a magical forest that is inhabited by a special species of creatures known as Fibonacci creatures. These creatures have a unique behavior where each creature's size is determined by the sum of the sizes of the previous two creatures. i.e. 1, 1, 2, 3, 5, 8, 13, 21, ...

The first two creatures in the forest have sizes 1 and 1. You want to determine the size of the**nth** Fibonacci creature in the forest.

Write a program that takes the value of **n** as input and calculates and returns the size of the**nth** Fibonacci creature using recursion.

**Function Specifications:**unsigned long long fibonacci(int num);

**Note:**This is a sample question asked in Capgemini recruitment.

**Input Format**

The input consists of the value of **n**.

**Output Format**

The output prints the nth Fibonacci term.

**Constraints**

1 <= n <= 30

**Sample InputSample Output**

10

55

**Sample InputSample Output**

20

6765

**Sample InputSample Output**

1

1

**Sample InputSample Output**

5

5

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3.

**Problem Statement**

Suppose you are working on a project that involves analyzing the individual digits of a given number. You are tasked with writing a program that calculates the sum of the digits of a positive integer using recursion.

Your program should take a positive integer as input and recursively calculate the sum of its digits. Finally, it should return the sum as the output.

**Function Specifications:**

int sumOfDigits(int num);

**Note:**This is a sample question asked in CTS recruitment.

**Input Format**

The input consists of a number **N**.

**Output Format**

The output prints the sum of the digits of the given number.

**Constraints**

0 < N <= 1000000

**Sample InputSample Output**

8659

28

**Sample InputSample Output**

52528

22

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4.

**Problem statement**

Emily is a mathematics enthusiast who loves exploring the world of numbers. Recently, she decided to create a program to calculate the factorial of a given number. She implemented a function called **factorial()** that takes an integer as an argument and uses the concept of **call by value** to calculate the factorial.

Whenever Emily calls the factorial function with a specific number, it performs the necessary calculations to find the factorial of that number.

Emily's program allows users to easily find the factorial of any number by simply providing the number as an argument to the factorial function. This way, they can quickly obtain the result without needing to implement complex factorial calculations themselves.

**Note:**This is a sample question asked in a TCS interview.

**Input Format**

The input consists of the number whose factorial has to be calculated.

**Output Format**

The output prints the factorial of the given number

**Sample InputSample Output**

7

5040

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5.

**Problem Statement**

Alex is a math enthusiast who loves playing with numbers. One day, Alex came across a challenge to write a program that reverses a given number. Intrigued by the challenge, Alex decides to write a program using reference variables.

Can you help Alex by creating a program that reverses the number using reference variables?

**Function Name:**void reverseNumber

**Note:** This is a sample question asked in a HCL interview.

**Input Format**

The input consists of a positive integer N.

**Output Format**

The output displays the reversed number N.

**Constraints**

0 <= N <= 107

Beginning and trialing zeros will be ignored in the number while reversing.

**Sample InputSample Output**

157

751

**Sample InputSample Output**

120

21

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6.

**Problem Statement**

Sara is a mathematician who loves exploring the properties of numbers. She recently wrote a program to calculate and print the cube of a given number. To accomplish this, she created a function **cube** by using **call by reference**.

Whenever someone calls Sara's cube function with a number, it calculates the cube of that number and directly modifies the original number. This means that the result is stored in the same variable that was passed as an argument.

Sara's program is designed to provide a convenient way to calculate and store the cube of a number without the need for additional variables or assignments. By utilizing the concept of call by reference, her program enables users to easily access the cube of any given number simply by calling the cube function

**Function Specification:**

cube(int& n)

**Note:** This is a sample question asked in a TCS interview.

**Input Format**

The input consists of an integer.

**Output Format**

The output prints the cube of the given number.

**Constraints**

The input integer values can be any integer in the range [1, 1000].

**Sample InputSample Output**

5

125

**Sample InputSample Output**

9

729

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7.

**Problem Statement**

Emma is an avid mathematician and wants to develop a program that can generate prime numbers within a given range. She believes that this program will be helpful for her research work.

Can you assist Emma by creating a C++ program that takes a starting and ending number as input and generates all the prime numbers within that range using a reference variable?

**Function Name:** void generatePrimesInRange

**Note:**This is a sample question asked in TCS recruitment.

**Input Format**

The input consists of start and end values separated by a space.

**Output Format**

The output displays the prime numbers in the given range separated by space.

**Constraints**

0 <= M, N <= 500

Start and end values are inclusive.

**Sample InputSample Output**

0 10

2 3 5 7

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8.

**Problem statement**

Write a program that takes three integer values as input from the user, then performs the following operations:

Calls a function named **shift** that takes three integer references as parameters and shifts their values one position to the left, i.e., the value of c becomes the value of b, the value of b becomes the value of a, and the value of a becomes the original value of c.

Calls a function named **getLargest** that takes three integer references as parameters and returns a reference to the largest value among them.

Prompts the user to input a new value for the largest value among the three integers, and modifies it using the reference returned by getLargest.Prints out the updated values of x, y, and z, as well as the new value of the largest value among them.

**Function Specifications:**

shift(int& a, int& b, int& c)

int& getLargest(int& a, int& b, int& c)

**Note:**This is a sample question asked in TCS recruitment.

**Input Format**

The first line consists of three integers x, y, and z separated by space representing the initial values of the variables.

The last line consists of an integer that represents the value to be assigned to the largest variable.

**Output Format**

The first line prints the original values of the variables before the shift operation.

The second line prints the values of the variables after the shift operation.

The third line prompts the values of the variables after the largest variable has been modified.

The fourth line prints the largest value among the three variables.

**Sample InputSample Output**

1 2 3

10

Before Shift: x: 1 y: 2 z: 3

After Shift: x: 3 y: 1 z: 2

After modification: x: 10 y: 1 z: 2

The largest value among x, y, and z is: 10

**Sample InputSample Output**

5 3 7

8

Before Shift: x: 5 y: 3 z: 7

After Shift: x: 7 y: 5 z: 3

After modification: x: 8 y: 5 z: 3

The largest value among x, y, and z is: 8

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9.

**Problem Statement**

Alice, a math enthusiast, wants to create a program that calculates the power of a given number. She decides to implement a function called **"power()"**that takes in two parameters: the base number (x) and the power (p). However, Alice wants to utilize call-by-reference to perform the calculation.

Alice's program prompts the user to enter the base number and the power they wish to raise it. Once the inputs are provided, the "power()" function is called with the memory addresses of the base number and the power variables. The function performs the necessary calculations and updates the value at the memory address of the base number variable.

Finally, Alice's program displays the result, which is the value of the base number raised to the power specified by the user.

**Note:**This is a sample question asked in Infosys recruitment.

**Input Format**

The input consists of the base and exponent.

**Output Format**

The output prints the result.

**Sample InputSample Output**

5 3

125

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10.

**Problem Statement**

Sarah is a math enthusiast who loves to analyze sets of numbers and calculate their sum. She recently wrote a program to find the sum of 'n' numbers using a function called **sum()**. To test her function, she created a program that allows her to input a set of numbers and compute their sum using the concept of **call by Value**.

When Sarah runs her program, she is prompted to enter the numbers one by one. After inputting all the numbers, the program invokes the sum() function. Inside the function, the numbers are passed by value, meaning that the original values are not modified. The function iterates through the provided numbers, accumulating their sum.

Once the computation is complete, the program displays the sum on the screen, allowing Sarah to quickly analyze the set of numbers she inputted. This Call by Value approach ensures that the original numbers remain unchanged, providing Sarah with accurate results and preserving the integrity of the input data.

**Note:**This is a sample question asked in Capgemini recruitment.

**Input Format**

The first line of the input consists of the value of n.

The next n lines of the input consist of the array of elements.

**Output Format**

The output prints the sum of elements.

**Sample InputSample Output**

5

14

25

36

47

58

180

**Sample InputSample Output**

8

123

456

789

147

258

369

951

753

3846

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q11.

**Problem Statement**

Sarah is studying number theory and is curious about the sum of divisors of a given number. She wants to write a C++ program to calculate this.

Can you help her in writing a program that takes a positive integer as input and calculates the sum of all its divisors using a reference variable?

**Function Name:** void calculateDivisorSum

**Note:**This is a sample question asked in Infosys recruitment.

**Input Format**

The input consists of an integer N.

**Output Format**

The output displays the sum of all the divisors of N.

**Constraints**

1 <= N <= 500

**Sample InputSample Output**

5

6

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q12.

**Problem Statement**

Alice is a mathematics teacher who wants to help her students understand the concept of the summation of natural numbers. To make it more interactive, she decides to write a program that calculates the sum of all natural numbers between 1 and a given number n using the concept **call by value**.

Alice's program implements a function called **calculatesum()**, which takes n as an argument. When the program runs, it prompts the user to enter a value for n. After receiving the input, the program calls the calculatesum() function, which performs the necessary calculations to find the sum of all natural numbers from 1 to n.

Alice's program provides an engaging way for her students to visualize and understand the concept of summation while also enhancing their programming skills.

**Note:**This is a sample question asked in TCS recruitment.

**Input Format**

The first line of input should be a positive integer representing the lower limit.

The second line of input should be a positive integer representing the higher limit.

**Output Format**

The output prints the sum of all natural numbers from 1 to n.

**Sample InputSample Output**

1

10

55

**Sample InputSample Output**

15

20

105

Time Limit: - ms Memory Limit: - kb Code Size: - kb