

# Question 1

**CP** - Rewrite the program 10 to find the sum until the user enters 0 or a negative number using **while** loop and break statement

**Hint =>**

1. Use infinite while loop as in **while (true)**
2. Take the user entry and check if the user entered 0 or a negative number to break the loop using **break;**

# Question 2

**CP** - Write a program to check for the natural number and write the sum of n natural numbers

**Hint =>**

1. A Natural Number is a positive integer (1,2,3, etc) sometimes with the inclusion of 0
2. A sum of n natural numbers is  $n * (n+1) / 2$

**I/P =>** number

**O/P =>**

If the number is a positive integer then the output is

The sum of \_\_\_ natural numbers is \_\_\_

Otherwise the output is

The number \_\_\_ is not a natural number

# Question 3

**CP** - Write a program to find the sum of numbers until the user enters 0

**Hint =>**

1. Create a variable total of type double initialize to 0.0. Also, create a variable to store the double value the user enters
2. Use the **while** loop to check if the user entered is 0
3. If the user entered value is not 0 then inside the while block add user entered value to the total and ask the user to input again
4. The loop will continue till the user enters zero and outside the loop display the total value

# Question 4

**CP** - Write a program to find the sum of n natural numbers using for loop, compare the result with the formulae  $n*(n+1)/2$  and show the result from both computations was correct.

**Hint =>**

1. Take the user input number and check whether it's a Natural number
2. If it's a natural number Compute using formulae as well as compute using **for** loop
3. Compare the two results and print the result

# Question 5

**CP** - Write a program to check whether a number is positive, negative, or zero.

**Hint =>**

1. Get integer input from the user and store it in the number variable.
2. If the number is positive, print positive.
3. If the number is negative, print negative.
4. If the number is zero, print zero.

# Question 6

**CP** - Write a program to count down the number from the user input value to 1 using a **while** loop for a rocket launch

**Hint =>**

1. Create a variable counter to take user inputted value for the countdown.
2. Use the **while** loop to check if the counter is 1
3. Inside a **while** loop, print the value of the counter and decrement the counter.

# Question 7

**CP** - Write a Program to find the factorial of an integer entered by the user.

**Hint =>**

1. For example, the factorial of 4 is  $1 * 2 * 3 * 4$  which is 24.

2. Take an integer input from the user and assign it to the variable. Check the user has entered a positive integer.
3. Using a **while** loop, compute the factorial.
4. Print the factorial at the end.

## Question 8

**CP** - Write a program to check if the first, second, or third number is the largest of the three.

**I/P** => number1, number2, number3

**O/P** =>

Is the first number the largest? \_\_\_\_

Is the second number the largest? \_\_\_\_

Is the third number the largest? \_\_\_\_

## Question 9

**CP** - Write a program SpringSeason that takes two int values month and day from the command line and prints "Its a Spring Season" otherwise prints "Not a Spring Season".

**Hint** =>

1. Spring Season is from March 20 to June 20

## Question 10

**CP** - Write a program to find the sum of n natural numbers using **while** loop compare the result with the formulae  $n*(n+1)/2$  and show the result from both computations was correct.

**Hint** =>

1. Take the user input number and check whether it's a Natural number
2. If it's a natural number Compute using formulae as well as compute using **while** loop
3. Compare the two results and print the result

## Question 11

**CP** - Write a program to check if the first is the smallest of the 3 numbers.

**I/P** => number1, number2, number3

**O/P** => Is the first number the smallest? \_\_\_\_

## Question 12

**CP** - Write a Program to find the factorial of an integer entered by the user.

**Hint** =>

1. For example, the factorial of 4 is  $1 * 2 * 3 * 4$  which is 24.
2. Take an integer input from the user and assign it to the variable. Check the user has entered a positive integer.
3. Using a **for** loop, compute the factorial.
4. Print the factorial at the end.

## Question 13

**CP** - Write a program to check if a number is divisible by 5

**I/P** => number

**O/P** => Is the number \_\_\_\_ divisible by 5? \_\_\_\_

## Question 14

**CP** - Write a program to check whether a person can vote, depending on whether his/her age is greater than or equal to 18.

**Hint** =>

1. Get integer input from the user and store it in the age variable.
2. If the person is 18 or older, print "The person can vote." Otherwise, print "The person cannot vote."

**I/P** => age

**O/P** =>

If the person's age is greater or equal to 18 then the output is

The person's age is \_\_\_\_ and can vote.

Otherwise

The person's age is \_\_\_\_ and cannot vote.

## Question 15

**CP** - Write a program to count down the number from the user input value to 1 using a **for** loop for a rocket launch

**Hint =>**

1. Create a variable counter to take user inputted value for the countdown.
2. Use the **for** loop to loop through the counter from end to the beginning
3. Inside a **for** loop, print the value of the counter

## Question 16

**CP** - Create a program to print the greatest factor of a number beside itself using a loop.

**Hint =>**

1. Get an integer input and assign it to the number variable. As well as define a greatestFactor variable and assign it to 1
2. Create a **for** loop that runs from last but one till 1 as in  $i = \text{number} - 1$  to  $i = 1$ .
3. Inside the loop, check if the number is perfectly divisible by  $i$  then assign  $i$  to greatestFactor variable and break the loop.
4. Display the greatestFactor variable outside the loop

## Question 17

**CP** - Write a program FizzBuzz, take a number as user input, and check for a positive integer. If positive integer, loop and print the number, but for multiples of 3 print "Fizz" instead of the number, for multiples of 5 print "Buzz", and for multiples of both print "FizzBuzz".

**Hint =>**

1. Take the user input number and check if it is a positive integer
2. Use **while** loop to display the output

## Question 18

**CP** - Create a program to print the greatest factor of a number beside itself using a loop.

**Hint =>**

1. Get an integer input and assign it to the number variable. As well as define the greatestFactor variable and assign it to 1
2. Create a variable counter and assign **counter = number - 1**; Use the **while** loop till the counter is equal to 1.
3. Inside the loop, check if the number is perfectly divisible by the counter then assign the counter to greatestFactor variable and break the loop.
4. Display the greatestFactor variable outside the loop

## Question 19

**CP** - Create a program to find the youngest friends among 3 Amar, Akbar, and Anthony based on their ages and the tallest among the friends based on their heights

**Hint =>**

1. Take user input for the age and height of the 3 friends and store it in a variable
2. Find the smallest of the 3 ages to find the youngest friend and display it
3. Find the largest of the 3 heights to find the tallest friend and display it

## Question 20

**CP** - Create a program to find all the multiples of a number taken as user input below 100.

**Hint =>**

1. Get the input value for a variable named number. Check the number is a positive integer and less than 100.
2. Create a counter variable and assign **counter = number - 1**; Use a **while** till the counter is > 1
3. Inside the loop, check if the counter perfectly divides the number.
4. If true, print the number and continue the loop.

## Question 21

**CP** - Create a program to find the factors of a number taken as user input.

**Hint =>**

1. Get the input value for a variable named number. Check if the number entered is a positive integer.
2. Create a counter variable and run the **while** loop till the counter is less than the user input number. In each iteration of the loop, check if the number is perfectly divisible by the counter. If true, print the value of the counter.

## Question 22

**CP** - Create a program to find the power of a number.

**Hint =>**

1. Get integer input for two variables named number and power.
2. Create a result variable with an initial value of 1.
3. Use the **for** loop from  $i = 1$  to  $i \leq \text{power}$ .
4. In each iteration of the loop, multiply the result with the number and assign the value to the result.
5. Finally, print the result

## Question 23

**CP** - Create a program to print odd and even numbers between 1 to the number entered by the user.

**Hint =>**

1. Get an integer input from the user, assign it to a variable number and check for Natural Number
2. Using a for loop, iterate from 1 to the number
3. In each iteration of the loop, print the number as odd or even number

## Question 24

**CP** - Create a program to find the factors of a number taken as user input.

**Hint =>**

1. Get the input value for a variable named number. Check if the number entered is a positive integer.
2. Run a **for** loop from  $i = 1$  to  $i < \text{number}$ . In each iteration of the loop, check if the number is perfectly divisible by  $i$ . If true, print the value of  $i$ .

## Question 25

**CP** - Create a program to find the bonus of employees based on their years of service.

**Hint =>**

1. Zara decided to give a bonus of 5% to employees whose year of service is more than 5 years.

2. Take salary and year of service in the year as input.
3. Print the bonus amount.

## Question 26

**CP** - Create a program to find all the multiple of a number taken as user input below 100.

**Hint =>**

1. Get the input value for a variable named number. Check the number is a positive integer and less than 100.
2. Use a **for** loop backwards: from i = 100 to i = 1.
3. Inside the loop, check if i perfectly divide the number.
4. If true, print the number and continue the loop.

## Question 27

**CP** - Create a program to find the multiplication table of a number entered by the user from 6 to 9.

**Hint =>**

1. Take integer input and store it in the variable number
2. Using a for loop, find the multiplication table of number from 6 to 9 and print it in the format number \* i =  
—

## Question 28

**CP** - Write a program FizzBuzz, take a number as user input, and check for a positive integer. If positive integer, loop and print the number, but for multiples of 3 print "Fizz" instead of the number, for multiples of 5 print "Buzz", and for multiples of both print "FizzBuzz".

*\*Hint =>*

1. Take the user input number and check if it is a positive integer
2. Use **for** loop to display the output

## Question 29

**CP** - Create a program to find the power of a number.



**Hint =>**

1. Get integer input for two variables named number and power.
2. Create a result variable with an initial value of 1.
3. Create a temp variable counter and initialize to zero. Use the **while** loop till **counter == power**.
4. In each iteration of the loop, multiply the result with the number and assign the value to the result. Also increment the counter.
5. Finally, print the result

## Question 30

**CP** - Write a LeapYear program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year.

**Hint =>**

1. The LeapYear program only works for year  $\geq 1582$ , corresponding to a year in the Gregorian calendar. So ensure to check for the same.
2. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.
3. Write code having multiple if else statements based on conditions provided above

## Question 31

**CP** - Write a Program to check if the given number is a prime number or not

**Hint =>**

1. A number that can be divided exactly only by itself and 1 are Prime Numbers,
2. Prime Numbers checks are done for numbers greater than 1
3. Loop through all the numbers from 2 to the user input number and check if the remainder is zero. If the remainder is zero break out from the loop as the number is divisible by some other number and is not a prime number.
4. Use the isPrime boolean variable to store the result and finally display the result

## Question 32

**CP** - Create a program to check if a number is Armstrong or not. Use the hints to show the steps clearly in the code

**Hint =>**

1. Armstrong Number is a number whose Sum of cubes of each digit results in the original number e.g.  
 $153 = 1^3 + 5^3 + 3^3$
2. Get an integer input and store it in the number variable, define sum variable, initialize it to zero and originalNumber variable, and assign it to the input number variable
3. Use the while loop till the originalNumber is not equal to zero
4. In the while loop find each digit which is the remainder of modulus operation ***number % 10***. Find the cube of the remainder and add it to the sum variable
5. Again in the while loop find the quotient of the number using division operation ***number / 10*** and assign it to the original number. This removes the last digit of the original number.
6. Finally check if the user number and the sum are the same, if the same it's an Armstrong number else not. So display accordingly

## Question 33

**CP** - Write a program to create a calculator using switch...case.

**Hint =>**

1. Create two double variables named first and second and a String variable named op.
2. Get input values for all variables.
3. The input for the operator can only be one of the four values: "+", "-", "\*" or "/".
4. Run a for loop from  $i = 1$  to  $i < \text{number}$ .
5. Based on the input value of the op, perform specific operations using the switch...case statement and print the result.
6. If op is +, perform addition between first and second; if it is -, perform subtraction, and so on.
7. If op is neither of those 4 values, print Invalid Operator.

## Question 34

**CP** - Create a program to find the BMI of a person

**Hint =>**

1. Take user input in double for the weight (in kg) of the person and height (in cm) for the person and store it in the corresponding variable.
2. Use the formula  $\text{BMI} = \text{weight} / (\text{height} * \text{height})$ . Note unit is  $\text{kg/m}^2$ . For this convert cm to meter
3. Use the table to determine the weight status of the person

BMI	Status
$\leq 18.4$	Underweight
18.5 - 24.9	Normal
25.0 - 39.9	Overweight
$\geq 40.0$	Obese

## Question 35

**CP** - Create a program to check if a number is an Abundant Number.

**Hint =>**

1. An abundant number is an integer in which the sum of all the divisors of the number is greater than the number itself. For example,  
 Divisor of 12: 1, 2, 3, 4, 6  
 Sum of divisor:  $1 + 2 + 3 + 4 + 6 = 16 > 12$
2. Get an integer input for the number variable.
3. Create an integer variable sum with an initial value of 0.
4. Run a for loop from  $i = 1$  to  $i < \text{number}$ .
5. Inside the loop, check if the number is divisible by  $i$ . If true, add  $i$  to the sum.
6. Outside the loop Check if the sum is greater than the number.
7. If the sum is greater than the number, print the Abundant Number. Otherwise, print Not an Abundant Number.

## Question 36

**CP** - Write a LeapYear program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year.

**Hint =>**

1. The LeapYear program only works for year  $\geq 1582$ , corresponding to a year in the Gregorian calendar. So ensure to check for the same.

2. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.
3. Write code with single if condition using logical and && and or || operators to meet the above conditions

## Question 37

**CP** - Create a program to check if a number taken from the user is a Harshad Number.

**Hint =>**

1. A Harshad number is an integer that is divisible by the sum of its digits. For example, 21 which is perfectly divided by 3 (sum of digits: 2 + 1).
2. Get an integer input for the number variable.
3. Create an integer variable sum with an initial value of 0.
4. Create a while loop to access each digit of the number.
5. Inside the loop, add each digit of the number to the sum.
6. Check if the number is perfectly divisible by the sum.
7. If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

## Question 38

**CP** - Write a program DayOfWeek that takes a date as input and prints the day of the week that the date falls on.

**Hint =>**

1. The program should take three command-line arguments: m (month), d (day), and y (year). For m use 1 for January, 2 for February, and so forth.
2. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth.
3. Use the following formulas, for the Gregorian calendar (where / denotes integer division):

$$y_0 = y - (14 - m) / 12$$

$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 \times ((14 - m) / 12) - 2$$

$$d_0 = (d + x + 31m_0 / 12) \bmod 7$$

# Question 39

**CP** - Create a program to count the number of digits in an integer.

**Hint =>**

- 1. Get an integer input for the number variable.
- 2. Create an integer variable count with the value 0.
- 3. Use a loop to iterate until the number is not equal to 0.
- 4. Remove the last digit from the number in each iteration
- 5. Increase the count by 1 in each iteration.
- 6. Finally, display the count to show the number of digits

# Question 40

**CP** - Write a program to input marks and 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as per the following guidelines

Grade	Remarks	Marks
A	(Level 4, above agency-normalized standards)	80% and above
B	(Level 3, at agency-normalized standards)	70-79%
C	(Level 2, below, but approaching agency-normalized standards)	60-69%
D	(Level 1, well below agency-normalized standards)	50-59%
E	(Level 1- , too below agency-normalized standards)	40-49%
R	(Remedial standards)	39% and below

**Hint =>**

- 1. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks