



Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

- (a) Write a program that takes an integer input from the user and uses the **ternary operator** to determine whether the number is **Odd** or **Even**. Display the appropriate message accordingly. (3)
- (b) Take  $n$  integers as input from the user. After each entry, print the current maximum value seen so far. Perform the entire process using a single loop.

**Example:** For  $n = 5$  and inputs:

3, 7, 4, 9, 6

The output should be:

- After 1st input: 3  
After 2nd input: 7  
After 3rd input: 7  
After 4th input: 9  
After 5th input: 9

(4)

- (c) Write a program that:

- Reads  $n$  integers from the user, each representing a day of the week (1 for Monday, 2 for Tuesday, ..., 7 for Sunday).
- As soon as the user enters a day of the week print corresponding day
- Uses a **for** loop to process each input.
- Inside the loop, uses a **switch** statement to print the corresponding day name.
- If the number is not in the range 1–7, print "Invalid day".

**Example:** For  $n = 5$  and input {1, 3, 7, 0, 5}, the output should be:

Monday  
Wednesday  
Sunday  
Invalid day  
Friday

(6)

- (d) For a given  $n$ , print a square pattern where:

- Odd rows contain the hash symbol #.
- Even rows contain the star symbol \*.



Department of Scientific Computing, Modeling and Simulation

**SC: 504 - Computational Lab - I**

Test - 1

Sem - I, M.Sc.(Scientific Computing)

Roll-No & Name : MS2503 DIVATE VEDANT .

Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

- (a) Write a program that accepts a single character representing a color code ('R', 'G', 'B', etc.) and displays a mood based on the selected color:

- 'R' or 'r': Angry
- 'G' or 'g': Happy
- 'B' or 'b': Sad

Use a **switch** statement to implement the logic and ensure **case-insensitivity**. Provide a default message for any unrecognized color code.

(3)

- (b) Accept integer  $n$  from user. Count down from  $n$  to 1, but skip all numbers ending in 4. Use one loop and continue.

(4)

- (c) Write a program that:

- Reads  $n$  integers from the user.
- Uses a **for** loop to process each number.
- Uses the **ternary operator** to print:
  - "Even" if the number is divisible by 2.
  - "Odd" otherwise.

**Example:** For  $n = 5$  and numbers {12, 7, 9, 24, 15}, the output should be:

Even  
Odd  
Odd  
Even  
Odd

(6)

- (d) For a given  $n$ , print a square pattern where:

- Odd rows contain the hash symbol #.
- Even rows contain the star symbol \*.

Use nested **while** loops to generate the pattern.

**Example:** For  $n = 4$ , the output should be:

```
# # # #
* * * *
# # # #
* * * *
```

(7)

— I would tell you a joke about UDP... but you might not get it. —



Department of Scientific Computing, Modeling and Simulation

**SC: 504 - Computational Lab - I**

Test - 1

Sem - I, M.Sc.(Scientific Computing)

Roll-No & Name : MS2508 JADHAV ANIKET .

Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

(a) You are given two integer inputs:

- `has_horn` (1 or 0)
- `can_fly` (1 or 0)

Based on the input values, classify the animal using logical and nested if conditions:

- If `has_horn` = 1 and `can_fly` = 0, print: "Unicorn"
- If `has_horn` = 0 and `can_fly` = 1, print: "Pegasus"
- If `has_horn` = 1 and `can_fly` = 1, print: "Alicorn"
- If `has_horn` = 0 and `can_fly` = 0, print: "Horse"

(3)

(b) Continuously take integer inputs from the user until the user enters -1. After each input, print the *rolling sum* (the cumulative total so far). If at any point the rolling sum becomes divisible by 13, reset the sum to 0.

**Example:** Input: 5  $\Rightarrow$  Sum = 5

Input: 4  $\Rightarrow$  Sum = 9

Input: 12  $\Rightarrow$  Sum = 21

Input: 4  $\Rightarrow$  Sum = 25

Input: 2  $\Rightarrow$  Sum = 27 (divisible by 13)  $\Rightarrow$  reset to 0

Input: 6  $\Rightarrow$  Sum = 6

Input: -1  $\Rightarrow$  End program.

(4)

(c) Write a program that:

- Reads the daily temperature for 7 days.
- Uses a for loop to process each day's temperature.
- Inside the loop, uses a nested if{else if structure to classify:
  - If temperature  $\geq$  35, print "Very Hot".
  - Else if temperature  $\geq$  25, print "Warm".
  - Else if temperature  $\geq$  15, print "Cool".
  - Else, print "Cold".

**Example:** For temperatures {36, 28, 19, 14, 37, 22, 30}, the output should be:

Very Hot

Warm

Cool

Cold

Very Hot

Cool

Warm

(6)

(d) Print multiplication tables from 1 to  $n$  in a grid format, where:

- Use an `outer for` loop to iterate through each table number.
- Use an `inner while` loop to print products from 1 to 10 for that table.
- If a product is divisible by 5, display it inside square brackets [ ] instead of plain number.

**Example:** For  $n = 3$ , the output should be:

1	2	3	4	[5]	6	7	8	9	[10]
2	4	6	8	[10]	12	14	16	18	[20]
3	6	9	12	[15]	18	21	24	27	[30]

(7)

— I would tell you a joke about UDP... but you might not get it. —



Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

- (a) Write a program that takes a student's score (ranging from 0 to 100) as input and uses the ternary operator to:

- Print "Pass" if the score is 40 or above, otherwise print "Fail".
- Additionally, display a comment using another ternary operator:
  - If the score is 90 or more, print "Excellent!"
  - Else, print "Keep improving!"

Ensure that the program handles valid input and outputs both the result and the comment accordingly.

(3)

- (b) Print the sum of all positive numbers in the Fibonacci sequence less than a given  $n$  that are divisible by 5. Use a while loop.

**Example:** For  $n = 50$ , the Fibonacci sequence below  $n$  is:

1, 1, 2, 3, 5, 8, 13, 21, 34

Numbers divisible by 5 are 5 only. Thus, the sum is:

5

For  $n = 200$ , the Fibonacci sequence below  $n$  is:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

Numbers divisible by 5 are 5 and 55. Thus, the sum is:

$$5 + 55 = 60$$

(4)

- (c) Write a program to read two integers (start, end) and classify each number in the range according to these rules:

- Use a single for loop to iterate through the numbers.
- Inside the loop, use a nested if{else if structure:
  - If the number is divisible by both 3 and 5, print "FizzBuzz".
  - Else if the number is divisible by 3, print "Fizz".
  - Else if the number is divisible by 5, print "Buzz".
  - Else, print the number itself.

**Example:** For start = 10, end = 16, the output should be:

Buzz  
11  
Fizz  
13  
14  
FizzBuzz  
16

(6)

(d) Print an  $n \times n$  checkerboard where:

- Use an outer `while` loop for rows.
- Use an inner `for` loop for columns.
- If the sum of the row and column index is even, print the current number (incrementing each time).
- If the sum is odd, print - instead.

Example: For  $n = 4$ , the output should be:

1 - 2 -  
- 3 - 4  
5 - 6 -  
- 7 - 8

(7)

— I would tell you a joke about UDP... but you might not get it. —

Buzz  
11  
Fizz  
13  
14  
FizzBuzz  
16

(6)

(d) Print an  $n \times n$  checkerboard where:

- Use an **outer while** loop for rows.
- Use an **inner for** loop for columns.
- If the sum of the row and column index is even, print the current number (incrementing each time).
- If the sum is odd, print - instead.

**Example:** For  $n = 4$ , the output should be:

1 - 2 -  
- 3 - 4  
5 - 6 -  
- 7 - 8

(7)

— I would tell you a joke about UDP... but you might not get it. —



Department of Scientific Computing, Modeling and Simulation

**SC: 504 - Computational Lab - I**

Test - 1

Sem - I, M.Sc.(Scientific Computing)

Roll-No & Name : MS2517 MAHIND SANIKA .

Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

- (a) Write a program that takes a student's score (ranging from 0 to 100) as input and uses the **ternary operator** to:
- Print "Pass" if the score is 40 or above, otherwise print "Fail".
  - Additionally, display a comment using another ternary operator:
    - If the score is 90 or more, print "Excellent!"
    - Else, print "Keep improving!"

Ensure that the program handles valid input and outputs both the result and the comment accordingly.

(3)

- (b) Ask the user for a positive integer  $n$ . Find the largest integer  $k$  such that

$$k! \leq n.$$

**Example.** For  $n = 150$ :

$$1! = 1, \quad 2! = 2, \quad 3! = 6, \quad 4! = 24, \quad 5! = 120, \quad 6! = 720.$$

Since  $5! = 120 \leq 150$  and  $6! = 720 > 150$ , the answer is  $k = 5$ .

(4)

- (c) Write a program that:

- Reads  $n$  integers from the user.
- Uses a **for** loop to process each number.
- Uses the **ternary operator** to print:
  - "Even" if the number is divisible by 2.
  - "Odd" otherwise.

**Example:** For  $n = 5$  and numbers  $\{12, 7, 9, 24, 15\}$ , the output should be:

Even  
Odd  
Odd  
Even  
Odd

(6)

- (d) Print a right-angled triangle of consecutive numbers starting from 1 upwards, for a given  $n$  (number of rows). Use nested **for** loops.

**Example:** For  $n = 4$ , the output should be:

1  
2 3  
4 5 6  
7 8 9 10

(7)

— I would tell you a joke about UDP... but you might not get it. —



Time: 10:15 AM to 11:15 AM

Date: August 14, 2025

Max mark: 20

**1. Attempt ALL**

- (a) Accept a number from the user and check for divisibility conditions. Use nested if-else statements to implement the logic.

Display the output based on the following rules:

- If the number is divisible by both 3 and 5, print: "FizzBuzz Trap!"
- If the number is divisible by 3 only, print: "Fizz"
- If the number is divisible by 5 only, print: "Buzz"
- If the number is divisible by neither, print: "Safe"

(3)

- (b) Given an integer  $n$ , find the first prime number  $p \geq n$  such that the integer obtained by reversing the digits of  $p$  is also prime. For example,  $13 \rightarrow 31$  (both prime). Hint: Use nested loops — one for checking numbers, another for checking primality.

(4)

- (c) Write a program that:

- Reads  $n$  integers from the user, each representing a day of the week (1 for Monday, 2 for Tuesday, ..., 7 for Sunday).
- As soon as the user enters a day of the week print corresponding day
- Uses a for loop to process each input.
- Inside the loop, uses a switch statement to print the corresponding day name.
- If the number is not in the range 1-7, print "Invalid day".

**Example:** For  $n = 5$  and input  $\{1, 3, 7, 0, 5\}$ , the output should be:

Monday  
Wednesday  
Sunday  
Invalid day  
Friday

(6)

- (d) Write a program that:

- Asks the user for the number of rows ( $n$ ) and columns ( $m$ ).
- Prints a table of numbers from 1 to  $n \times m$  along with their squares.
- Uses an outer while loop to control the rows and an inner for loop to control the columns.

**Example Run:**

Daily Icadam.

Enter number of rows: 2  
Enter number of columns: 3

Number	Square
1	1
2	4
3	9
4	16
5	25
6	36

(7)

— I would tell you a joke about UDP... but you might not get it. —

**1. Attempt ALL**

- (a) Accept a number from the user and check for divisibility conditions. Use **nested if-else** statements to implement the logic.

Display the output based on the following rules:

- If the number is divisible by both 3 and 5, print: "FizzBuzz Trap!"
- If the number is divisible by 3 only, print: "Fizz"
- If the number is divisible by 5 only, print: "Buzz"
- If the number is divisible by neither, print: "Safe"

(3)

- (b) Given  $n$  and  $m$ , find the first number greater than  $n$  that is divisible by  $m$  using one loop.

(4)

- (c) Write a program that:

- Reads the marks of  $n$  students (integer values between 0 and 100).
- Uses a **for** loop to process each student's marks.
- Inside the loop, uses a **nested if{else if** structure to classify:
  - If marks  $\geq 90$ , print "Grade A".
  - Else if marks  $\geq 75$ , print "Grade B".
  - Else if marks  $\geq 60$ , print "Grade C".
  - Else if marks  $\geq 40$ , print "Grade D".
  - Else, print "Fail".

**Example:** For  $n = 5$  and marks  $\{95, 82, 61, 47, 30\}$ , the output should be:

Grade A

Grade B

Grade C

Grade D

Fail

(6)

- (d) Print an  $n \times n$  checkerboard where:

- Use an **outer while** loop for rows.
- Use an **inner for** loop for columns.
- If the sum of the row and column index is even, print the current number (increasing each time).
- If the sum is odd, print - instead.

**Example:** For  $n = 4$ , the output should be:

1 - 2 -  
- 3 - 4  
5 - 6 -  
- 7 - 8

(7)

— I would tell you a joke about UDP... but you might not get it. —

4  
1 2 2 3 2 1 2  
2 3 1 2 4  
3 1 2  
2

**1. Attempt ALL**

(a) You are given two integer inputs:

- `has_horn` (1 or 0)
- `can_fly` (1 or 0)

Based on the input values, classify the animal using logical and nested if conditions:

- If `has_horn` = 1 and `can_fly` = 0, print: "Unicorn"
- If `has_horn` = 0 and `can_fly` = 1, print: "Pegasus"
- If `has_horn` = 1 and `can_fly` = 1, print: "Alicorn"
- If `has_horn` = 0 and `can_fly` = 0, print: "Horse"

(3)

(b) Accept integer  $n$  from user. Count down from  $n$  to 1, but skip all numbers ending in 4. Use one loop and continue.

(4)

(c) Write a program that:

- Reads  $n$  integers from the user, each representing a day of the week (1 for Monday, 2 for Tuesday, ..., 7 for Sunday).
- As soon as the user enters a day of the week print corresponding day
- Uses a `for` loop to process each input.
- Inside the loop, uses a `switch` statement to print the corresponding day name.
- If the number is not in the range 1-7, print "Invalid day".

**Example:** For  $n = 5$  and input {1, 3, 7, 0, 5}, the output should be:

Monday  
Wednesday  
Sunday  
Invalid day  
Friday

(6)

(d) Print an  $n \times n$  checkerboard where:

- Use an outer `while` loop for rows.
- Use an inner `for` loop for columns.
- If the sum of the row and column index is even, print the current number (increasing each time).
- If the sum is odd, print - instead.

**Example:** For  $n = 4$ , the output should be:

1 2 3 4  
 0 1 2 3 - rows while  
 - 1 2 1 2 - 0 ¶  
 - 2 1 - 3 - 4  
 3 2 5 - 6 -  
 4 3 - 7 - 8  
 Colm Fer

(7)

— I would tell you a joke about UDP... but you might not get it. —

0	1	2	3	1 - 2 -
1	2	3	4	- 3 - 4
2	3	4	5	5 - 6 -
3	4	5	6	- 7 - 8

$i = 0, j = 0;$   
 $n = 4, \text{crn} = 0 = 0$

```

while (i <= n)
  crn++;
  for (j = 0; j <= n; j++)
  {
    if ((i + j) % 2 == 0)
      printf("1-d", nr);
      break;
    else
      printf("2-");
  }
}

```

{

i++;