SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya School of Engineering

K. J. Somaiya School of Engineering, Mumbai-77
(Somaiya Vidyavihar University)

Somaiya TRUST

Course:	Programming in C		**Semester:**	II	**Date:**	
Division:		**Batch:**	**SET**	C	**Name:**	
Exam:	OST		**Time:**		**Roll No:**	

<table>
<tr><th>Q 1</th><th>Attempt Any ONE [Show all test Cases in output.]</th><th>Marks</th></tr>
<tr><td>1</td><td>Write a C program to find and print the first N prime numbers using a for loop. Use the break statement to stop the loop when N prime numbers have been found. Test the program with the following test cases:
<table>
<tr><td><b>Test Case 1:</b>
N = 5
2 3 5 7 11</td><td><b>Test Case 2:</b>
N=8
2 3 5 7 11 13 17 19</td><td><b>Test Case 2:</b>
N=12
2 3 5 7 11 13 17 19 23 29 31 37</td></tr>
</table>
<b>Input from user 1 marks</b>
<b>check if a number is prime 3 marks</b>
<b>print first N prime numbers 3 marks</b>
<b>Check all test cases 1 marks</b></td><td>08</td></tr>
<tr><td>2</td><td>Write a C program to take an MxN matrix as input and find the <b>column</b> with the highest sum. Display the column number and the highest sum as output. Test the program with the following test cases:
<table>
<tr><td><b>Test Case 1:</b>
3 x 3 matrix:
1 2 3
4 5 6
7 8 9
Expected output: Column 3 has the highest sum = 18</td><td><b>Test Case 2:</b>
2 x 4 matrix:
3 4 2 1
5 6 7 8
Expected output: Column 2 has the highest sum = 10</td><td><b>Test Case 3:</b>
4 x 2 matrix:
1 2
3 4
5 6
7 8
Expected output: Column 2 has the highest sum = 20</td></tr>
</table>
<b>Input: Matrix size & its element from user 2 marks</b>
<b>find the column with the highest sum 4 marks</b>
<b>Display the column number and the highest sum for all test cases 2 marks</b></td><td>08</td></tr>
</table>

<table>
<tr><th>Q 2</th><th>Attempt Any ONE [Show all test Cases in output.]</th><th>Marks</th></tr>
<tr><td>1</td><td>Define a structure Student with fields <b><i>name, roll_number, and marks(an array of 5 subjects)</i></b>. Write functions to:
● Add a new student.
● Calculate the average marks of each student.
● Display student details with average marks.
Implement these functions and demonstrate their usage in a menu-driven program.</td><td>12</td></tr>
</table>

Test Case 1:	**Test Case 2:**	**Test Case 3:**
Add student: Name: John Roll Number: 101 Marks: 85, 90, 78, 88, 92 Expected output: John’s details displayed, Average Marks = 86.6	Add student: Name: Mary Roll Number: 102 Marks: 76, 82, 88, 91, 79 Expected output: Mary’s details displayed, Average Marks = 83.2	Add student: Name: Alex Roll Number: 103 Marks: 95, 87, 90, 86, 88 Expected output: Alex’s details displayed, Average Marks = 89.2

Define structure for student 2 marks
Function to add new student 3 marks
Function to calculate avg mark 3 marks
Function to display mark 2 marks
Implement Switch-case 2 marks

2 A N-digit positive integer is entered through the keyboard. Write a function to calculate the product of the digits of the N-digit number: 12
1.Without using recursion – use call by reference.
2.Using recursion.

Selection of type of function should be runtime.

Test Case 1:	**Test Case 2:**	**Test Case 3:**
Input: 12345 Expected output: Product of digits = 120	Input: 54321 Expected output: Product of digits = 120	Input: 11111 Expected output: Product of digits = 1

User selection runtime(if-else or switch-case) 2 marks
Function productWithoutRecursion 3 marks
Function productWithRecursion 3 marks
Display result with all testcase 1 marks
Use of Pointer 3 marks