



Daffodil International University
Faculty of Science & Information Technology
Department of Computer Science & Engineering
Mid Examination, Fall 2026

Course Code: CSE113, Course Title: Programming and Problem Solving

Level: 1 Term: 2 Batch: 72

Time: 01:30 Hrs

Marks: 25

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.	Analyze the following C program and answer the questions below- <pre> 1. #include <stdio.h> 2. int main() 3. { 4. int N=5, sum=0; 5. for(int i=1; i<=N; i++) 6. { 7. sum = sum + i; 8. } 9. printf("Sum = %d", sum); 10. }</pre>	CO1
a)	Classify all the tokens in the given C program into keywords, operators, constants, and identifiers into an organized structural table.	[2]
b)	Modify the program to find the factorial of the number N. [Factorial is a mathematical operation where you multiply a positive whole number by every whole number below it down to 1. Formula: $(n! = n * (n - 1) * (n - 2) \dots * 1)$]	[2]
2.	<pre> 1. #include <studio.h> 2. int main() 3. { 4. float total = 100.0; 5. int discount = 10; 6. int price = total - discount; 7. if (price = 90); 8. { 9. printf("ok.\n"); 10. } 11. else{ 12. Printf("Not ok.\n"); 13. } 14. return 0; 15. }</pre>	CO2
a)	Debug the given C program by correcting the coding errors, and justify each change based on standard C compilation.	[2]

b)	Write the corrected version of the above program.	[2]					
c)	Write down the output of the following program written in C. <pre>1. #include <stdio.h> 2. int main() 3. { 4. int scores[] = {55, 82, 45, 90, 71}; 5. int total_elements = 5; 6. int sum = 0; 7. int pass_count = 0; 8. printf("--- Student Results ---\n"); 9. for (int i = 0; i < total_elements; i++) 10. { 11. printf("Student %d scored: %d\n", i + 1, scores[i]); 12. sum = sum + scores[i]; 13. if (scores[i] >= 50) 14. { 15. pass_count++; 16. } 17. } 18. float average = (float)sum / total_elements; 19. printf("-----\n"); 20. printf("Average Score: %.1f\n", average); 21. printf("Total Students Passed: %d\n", pass_count); 22. return 0; 23.)</pre>	[2]	CO2				
3.	Construct a complete C program for each of the specific problem scenarios provided below		CO3				
a)	A factory wants to calculate how much water a cylindrical tank can hold. The tank's radius and height are measured in meters. Write a program that takes the radius <u>r</u> and height <u>h</u> as input and calculates the volume of the tank. The volume of a cylinder is calculated using the formula- $v = \pi r^2 h$ Use $\pi = 3.1416$ Input: - Two integers <u>r</u> and <u>h</u>, representing the radius and height of the cylindrical tank, respectively. Output: - Print the volume of the tank up to 3 decimal places. <table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>3 5</td><td>141.372</td></tr></table>	Sample Input	Sample Output	3 5	141.372	[3]	
Sample Input	Sample Output						
3 5	141.372						
b)	A laboratory is conducting an experiment to study the growth of a particular species of cells. Initially, there are <u>n</u> cells in the laboratory sample. The cells are observed to grow exponentially, such that after <u>t</u> time units, the total number of	[3]					

CO3

cells becomes n^t . Write a program that takes the initial number of cells n and the time period t as input and calculates the total number of cells after t time units.

N. B. You can't use a built-in standard library function.

Input

- The input consists of two integers, n ($1 \leq n \leq 10$) and t ($1 \leq t \leq 10$), representing the initial number of cells and the elapsed time period, respectively.

Output

- Print a single integer representing the total number of cells after t time units.

Sample Input	Sample Output
2 5	32

c) A gas distribution company calculates the monthly gas bill for its customers based on the amount of gas consumed. The company applies different rates per unit according to the monthly consumption range. Now you have to write a program to calculate and display the total monthly gas bill based on the following rates.

[3]

Gas Consumed (Unit)	Rate per Unit (BDT)
1 to 50	4
51 to 100	6
101 to 200	8
Above 200	12

Note:

- If the consumed units are less than or equal to 0, display "Invalid Input".
- A fixed service charge of 75 BDT is added to every valid monthly bill.

Input: A single integer representing the total gas consumption in units for the month.

Output: Display the total monthly gas bill in the format shown in the sample output.

Sample Input	Sample Output
150	Total bill = 1275 taka

d) A contest is held where students receive marks in the first round. Students who score greater than or equal to the average marks of all participants will qualify for the second round. Write a program to determine how many students will advance to the next round.

[3]

Input:

- First line contains an integer N representing the number of students in the class
- N space-separated integer numbers stored in an array representing the marks of students

Output: Average mark with two decimal number and total number of students qualifying for the next round

Sample Input	Sample Output
5 10 9 12 7 19	Average mark = 11.40 2 students

CO3

- e) Once upon a time, there was a digital kingdom named Las-Ak. The King of the kingdom decided to host a grand festival. He invited citizens from all over the land. As the guests arrived, the Royal Gatekeeper wrote down their unique ID numbers in a long scroll called a 1D Array. However, the Gatekeeper noticed a problem: some mischievous citizens had snuck into the line multiple times to get extra festival food coupons! The King ordered his Grand Programmer to find every ID that appeared more than once. Here is how the Grand Programmer solved the mystery using C. [3]

Input

- The first line contains a single integer, N , representing the total number of elements in the array.
- The subsequent lines contain N integers, representing the individual elements of the array.

Output

- If duplicates are present: Print each duplicate value separated by space.

Sample Input	Sample Output
10 10 20 10 30 20 50 20 40 90 40	10 20 40