



Reg. No.

Indian Institute of Information Technology, Surat
Department of Computer Science & Engineering
End Semester Examination 1st Sem, 2025-26
Fundamentals of Computer and C Programming (CS101)

Timing: 10:00 AM - 1:00 PM

Date: 1 Dec 2025

Max Marks: 100

Instructions: Attempt all parts of a question at one place. If you do not, later parts may not be graded. **This exam has 3 printed pages.** Please ensure that you have received all three.

1. State whether the following statements are true or false. If the statement is false then write the correct statement. (8x2=16)

- a. Writing comments in a C program increases its runtime.
- b. The format specification “%4s” prints the first 4 characters of the given string.
- c. In von Neumann architecture, there are separate memories for instructions and data.
- d. For an unsigned integer x , the expression $x\&111$ is identical to $x\&8$.
- e. Global variables are accessible by all functions of the program.
- f. Memory allocated by `malloc` inside a function gets freed when the function returns.
- g. `typedef int* IP; IP ptr` defines a variable `ptr` that is a pointer to integers.
- h. If `fopen()` fails to open a file, it returns the EOF (End-Of-File) macro.

2. Choose the single correct answer for each of the following questions. (8x2=16)

- a. What is copied into a local variable inside the function when an array is passed to a function as an argument?
 - i. Entire array
 - ii. The value of the first element of the array
 - iii. The pointer to the first element of the array
 - iv. The name of the array
- b. Nesting is allowed for which of the following statements?
 - i. if
 - ii. if-else
 - iii. switch
 - iv. All of these
- c. Which of the following is true for the C language?
 - i. Interpreted
 - ii. Compiled
 - iii. Low level language
 - iv. Case insensitive
- d. Consider an array declared as `int arr[10][10]`. Which of the following **does not** point to the base address of this array?
 - i. `arr`
 - ii. `arr[0]`
 - iii. `arr[0][0]`
 - iv. `&arr[0][0]`
- e. Which constant value can be assigned to a pointer variable?
 - i. 0
 - ii. NULL
 - iii. Both i & ii
 - iv. None of i or ii
- f. What is the return data type of the `malloc()` function?
 - i. `void*`
 - ii. `int*`
 - iii. `pointer`
 - iv. `struct`
- g. Which of the following is an exit-controlled loop?
 - i. `for`
 - ii. `while`
 - iii. `do-while`
 - iv. `goto`
- h. When you execute the compiled C program, where are the variables stored?
 - i. ROM
 - ii. RAM
 - iii. Hard Drive
 - iv. Terminal

3. Write the output for the C code snippets below. Disregard compilation warnings, if any. There is **no** runtime error or undefined behavior. Assume standard context and 64-bit Linux environment identical to lab computers. (4x2=8)

a. <pre>long x = 42; long *y = &x; int inc_x = (long)(x+1) - (long)x; int inc_y = (long)(y+1) - (long)y; printf("inc_x=%d ", inc_x); printf("inc_y=%d ", inc_y);</pre>	c. <pre>int arr1[5][20]; int* arr2 = (int*)malloc(sizeof(arr1)); printf("sz1=%d ", sizeof(arr1)); printf("sz2=%d ", sizeof(arr2));</pre>
b. <pre>int x = 10; int count = 0; count += (x==10 && x>10 && !x); count += (x==10 && x<10 !x); count += (x==10 && x>10 !x); count += (x==10 x>10 !x); printf("x=%d count=%d", x, count);</pre>	d. <pre>struct measurement { int val; char unit; }; struct measurement h = {1, 'm'}; struct measurement w = h; h.val = 2; printf("h=%d%c ", h.val, h.unit); printf("w=%d%c", w.val, w.unit);</pre>

4. For **any three** of the C code snippets below, classify it as compilation error, runtime error, or undefined behavior. Then, **fix and rewrite** the corrected code snippet assuming standard context. (3x5=15)

- `char wish[10] = "good luck!"; //String wish should store "good luck!"`
- `/* Strings prefix, code, and course should store "CS", "101", and "CS101" respectively. String course should be derived from prefix and code. */
char *prefix = "CS"; char code[] = "101";
char course[101] = prefix + code;`
- `int num1, num2; int* ptr1 = num1, ptr2 = num2; /* ptr1 and ptr2 should store pointers to num1 and num2 respectively. */`
- `int even_primes[1]; even_primes = {2}; /* even_primes is an array of length one and should store a single element: number 2. */`

5. Differentiate between **any three** of the following: (3x5=15)

- Function declaration and function definition. Give a code example where a function declaration is strictly necessary for the successful compilation.
- Explain the 'structure' declaration and their initialization in 'C'. Demonstrate through a coding example how the elements of a structure are accessed.
- 1's Complement and 2's Complement conventions to store negative integers. State the representation of -22 as a 32-bit signed integer using 2's complement notation.
- Pass-by-Value and Pass-by-Reference for function arguments. State in one line how changes made inside the function affect the actual arguments in the caller function.

6. Write complete C programs for **any two** of the following. You should properly indent and add comments in your code. **(2x15=30)**

- a. Read a line of text (string) from the user, followed by a single character to be searched for. You may assume the string will not exceed **1000** characters. Implement a function with the prototype: `void replace_char(char *str, char c);` It should use pointers to iterate through `str` and replace every occurrence of `c`, if any, with an `*`. The main function should then display the updated string.
- b. Take as input a **5x5** integer matrix as a 2D array from the user and determine if it is symmetric. Print **Symmetric** or **Not Symmetric** accordingly. Then, find the sum of all of its eigenvalues (with multiplicity) and print it.
- c. The Collatz sequence (also known as the **3n+1** problem) for any positive integer `n` is as follows: If `n` is **1**, the sequence ends. If `n` is even, the next term is `n/2`. If `n` is odd, the next term is **3n+1**. Implement a recursive function `void collatz(int n)` that takes input an integer `n` and prints the corresponding Collatz sequence. For example, the Collatz sequence for `n=10` is: **10 5 16 8 4 2 1**. You may assume that the input `n` is a positive **32-bit int** chosen such that the sequence is finite and its maximum term is less than 2^{31} . Write and call this recursive function `collatz()` from the `main` function with parameter `n=2025`.