

C Programming Practice Questions

Government Engineering College Kishanganj

Lalan Kumar Ram (Assistant Professor)

lalan.cse@gmail.com

Department of Computer Science and Engineering

C Programming Assignment

General Instructions

1. To run all C programs on a Linux system, ensure that the GCC compiler is installed by executing the following commands in the terminal:

```
sudo apt update
sudo apt install gcc
sudo apt install build-essential
gcc --version
```

2. To run a C program (suppose the program file name is "program.c"):

- (a) Open the terminal.
- (b) Compile the program using GCC:

```
gcc program.c -o program
```

- (c) After successful compilation, run the program using:

```
./program
```

3. For all questions, you are allowed to use only the following header files:

```
#include <stdio.h>
#include <math.h>
```

Do not use any other header file.

Q1. Armstrong number (N-digit)

Write a C program that reads an N -digit integer and checks whether it is an Armstrong number or not. For example,

$$153 = 1^3 + 5^3 + 3^3.$$

Q2. Alphabet case conversion

Write a C program that takes a single alphabet character as input.

- If the character is a lowercase letter, convert it to uppercase.
- Otherwise, print the character unchanged.

Q3. Name initial and year of birth

Write a C program that inputs:

- The first character of your name, and
- Your date of birth in the format DDMMYYYY (as an integer).

Then display the first letter of your name and the year of your birth in the following format.

Example:

Enter Your First Letter: L
Enter Your DOB: 01111950

Output:
First Letter: L
DOB Year: 1950

Q4. Roots of a quadratic equation

Write a C program to input the values of a , b , and c (coefficients of a quadratic equation) and compute the two roots of the quadratic equation

$$ax^2 + bx + c = 0.$$

Use functions from `math.h` wherever necessary for square root and power operations.

Q5. Simple calculator

Write a C program that works as a basic calculator to evaluate a single arithmetic expression.

Requirements:

- The input format should be: `value1 operator value2` (e.g., `2+3`, `10*5`).
- Supported operators:
 - Addition: `+`
 - Subtraction: `-`
 - Multiplication: `*`
 - Division: `/`
- The program must not print any extra messages. The output should contain only the numerical result.

Example:

Input: `2+3`
Output: `5`

Q6. Consecutive digit numbers from n

Write a C program that, given an integer n , prints all integers formed from its first digit, first two digits, and so on, up to all digits.

Example: If $n = 7658$, the output should be:

$$\{7, 76, 765, 7658\}.$$

Q7. Pattern printing

Write a C program to print the following pattern:

```
1A
12AB
123ABC
1234ABCD
```

Q8. Infinite series for $\sin(x)$ (Maclaurin series)

The Maclaurin series expansion for $\sin(x)$ is:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

For $x = 0.50$, write a C program to compute the sum of this series, adding terms until the absolute difference between two consecutive terms is less than 0.0001.

Q9. Cubic polynomial: roots, coefficients, point values

A degree-3 polynomial with roots r_0 , r_1 , and r_2 can be written as:

$$f(x) = c(x - r_0)(x - r_1)(x - r_2),$$

where c is a constant. In this question, assume $c = 1$.

Part A: Root to coefficient form

Write a C program that:

- Prompts the user to enter three integer roots: r_0 , r_1 , and r_2 .
- Computes the coefficients (a_3, a_2, a_1, a_0) of the polynomial in standard form:

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0.$$

- Displays the polynomial in coefficient form.

Part B: Point-value representation

Extend the program from Part A to:

- Prompt the user to enter four integer evaluation points: x_0, x_1, x_2, x_3 .
- For each point, first check whether it matches any of the roots (r_0, r_1, r_2) .
 - If any input point is equal to a root, print an error message and terminate the program.
- If none of the points are roots, evaluate $f(x)$ at each point and display the point-value representation as:

$$\{(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3))\}.$$

Q10. Reverse an N -digit number (two methods)

Write a C program to reverse an N -digit integer in two different ways:

- Using a loop with the modulo (%) operator.
- Using a loop without the modulo (%) operator.

Q11. Count digits without division or modulo

Write a C program to count the number of digits in an integer using a loop, without using division (/) or modulo (%) operators.

Array-based Questions

Q12. Half ascending, half descending

Write a C program to sort the first half of an array in ascending order and the second half in descending order.

Q13. Reverse array in-place

Write a C program to reverse an array without using another array.

Q14. Second largest and second smallest

Write a C program to find the second largest and second smallest elements in an array.

Q15. Frequency of each element

Write a C program to count and display the frequency of each element in an array.

Q16. Move all zeros to the end

Write a C program to shift all zeros in an array to the end, without changing the relative order of the non-zero elements.

Q17. Elements greater than average

Write a C program to compute the average of all elements in an array and count how many elements are greater than this average.

Q18. Palindromic array check

Write a C program to check whether a given array is a palindrome or not.

Q19. Right rotation by k positions

Write a C program to rotate an array to the right by k positions. The rotation should be performed in-place.

Q20. Two sorted arrays A and B

You are given two arrays $A[0..n-1]$ and $B[0..m-1]$ of positive integers such that $n > 0$ and $m > 0$. Both A and B are sorted in non-decreasing order.

Write a C program to process these arrays so that:

- After processing, both A and B remain sorted in non-decreasing order.
- No element in A is larger than any element in B.
- Elements may be moved between A and B to satisfy the above condition.
- You must not use any third array and may use only a constant number of extra integer variables.
- Assume $n, m \leq 100$.
- Your program should contain only the `main()` function.

Example:

$$\begin{aligned}n &= 3, & m &= 4, \\A &= \{1, 3, 15\}, \\B &= \{2, 9, 18, 26\}.\end{aligned}$$

A valid final configuration is:

$$A = \{1, 2, 3\}, \quad B = \{9, 15, 18, 26\}.$$

Q21. Count vowels and total characters

Write a C program to read a sequence of characters from the keyboard, terminated by the newline character '`\n`'. Count and display:

- The number of vowels entered.
- The total number of characters entered.

Q22. Reverse string without extra array

Write a C program to read a sequence of characters from the keyboard, terminated by the newline character '`\n`', and reverse the string in-place without using any extra character array.