FOCP 1 Assignment Submission Instructions (via GitHub)

To ensure uniformity and avoid confusion, please follow these steps carefully while submitting your assignments:

- Create a **folder on your GitHub repository** named "**FOCP1-Assignment**" **.**
- Save each assignment code file using the format **Q1.c, Q2.c, Q3.c**, etc., matching the question numbers.
- For **LeetCode** or **HackerEarth** problems (**Q19 to Q24**):
  - Take a **screenshot** (e.g. Q19.png/Q19.jpg) of your **submitted code** showing the **Accepted/Submitted** status.
  - Upload this screenshot **to the same folder** as the related code file.
- Make sure all files and screenshots are **clearly named and properly organised**.
- **Commit and push** the folder to your GitHub repository **before the submission deadline**.
- Check that your code **runs correctly** before uploading.

Assignment 1

Submission Date -2-Nov-2025

Q1. Develop a C program to check whether a given number is an Armstrong number or not, demonstrating understanding of number manipulation and digit-based computations used in digital verification systems.

Q2. Construct a C program to find the HCF (Highest Common Factor) of two integers using iterative or recursive logic — a core operation in optimisation algorithms and cryptographic computations.

Q3. Design a C program to subtract two integers without using the minus (-) operator, applying bitwise operators. Highlight how such logic can be used in low-level arithmetic operations in embedded systems or processors.

Q4. Implement a C program to swap two numbers using four different methods, demonstrating diverse approaches such as arithmetic, bitwise XOR, pointer manipulation, and temporary variable usage — a key skill in memory and variable management.

Q5. Develop a C program to determine the quadrant in which a given coordinate point (x, y) lies on the XY plane, illustrating the use of conditional statements and coordinate geometry logic relevant in game development and robotics navigation.

Q6. Create a C program that allows the user to convert between Binary and Decimal systems based on their choice, showcasing understanding of data encoding and computer number systems used in hardware-level design.

Q7. Develop a C program to print a binary pyramid pattern.

```
0          0

01         01

010       010

0101   0101

0101001010
```

Q8. *Develop a C program to generate the Fibonacci series up to 'n' terms, where each term is the sum of the two preceding ones.*

0, 1, 1, 2, 3, 5, 8, 13, ...

Assignment 2

Submission Date -02-Nov-2025

Q9. Design a C program to find the first occurrence of the score “99” in an array, focusing on linear search and data retrieval techniques.

Q10. Implement a program to find who and how many students scored “99” in the marks array, emphasising data scanning and frequency counting.

Q11. Develop a C program to traverse an array of scores, determine whether each score is even or odd using conditional logic, and store them into two separate arrays — even_array and odd_array.

Q12. Develop a C program to find the maximum and minimum scores in an array, applying comparative logic for ranking and analysis.

Q13. Design a C program to find a peak element that is not smaller than its neighbours.

Q14. Develop a C program to count the number of prime numbers in an array.

Q15. Write a C program to cyclically rotate the array clockwise by one position, applying array transformation logic used in scheduling and encryption.

Input: arr[] = {1, 2, 3, 4, 5}

Output: arr[] = {5, 1, 2, 3, 4}

Input: arr[] = {2, 3, 4, 5, 1}

Output: {1, 2, 3, 4, 5}

Q16. Implement a C program to insert an element at the front, middle, or end of an array, and print the array before and after insertion.

Q17. Design a C program to delete an element from the front, middle, or end of an array, and print the array before and after deletion.

Q18. Develop a program to identify and print duplicate elements in an array, or print “-1” if no duplicates exist, applying frequency detection and data validation.

Examples:

Input: {2, 10,10, 100, 2, 10, 11,2,11,2}

Output: 2 10 11

Input: {5, 40, 1, 40, 100000, 1, 5, 1}

Output: 5 40 1

Assignment 3

Submission Date -08-Nov-2025

Q19. **35. Search Insert Position**

Q20. **136. Single Number**

Assignment 4

Submission Date -15-Nov-2025

Q21. https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/practice-problems/algorithm/roy-and-profile-picture/ **(Roy and Profile Picture)**

Q22. https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/practice-problems/algorithm/is-zoo-f6f309e7/ (ZOO)

Q23. https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/practice-problems/algorithm/mojtaba-prepares-contest-29b2a044/ (Cost of Balloons)

Q24. https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/practice-problems/algorithm/modify-the-string/ (Toggle String)