

**Programming
Fundamentals Lab
(CL1002)**

Date: 17/10/2024

Course Instructor(s)

Muhammad Monis, *Shaheer Ahmad*

Lab Mid Exam (B)

Total Time: 90 minutes

Total Marks: 20

Total Questions: 03

Semester: FL-2024

Campus: Karachi

Dept: Computer Science)

Submission Instructions:

- You must comment your student ID on top of each file. (Line#1 of your code).
- Name the file for each question according to Roll_No e.g. k24-xxxx_Q1.c, k24-xxxx_Q2.c etc.
- Submission is via a client software so open the application present on the Desktop.

CLO # 1: Understand and Analyze flowcharts, PAC (Process-Activity-Control) charts, and IPO (Input-Process-Output) models to represent system workflows and these diagrams into algorithm and pseudocode implementation.

Q1. [5 marks] Convert the following Pac chart to C-code

Column	Details
Given Data (Inputs)	<ul style="list-style-type: none">- Item Price- Quantity of items added to the cart- Discount percentage (if applicable)
Required Results (Outputs)	<ul style="list-style-type: none">- Total Price of items in the cart
Processing	<ul style="list-style-type: none">- Multiply Item Price by Quantity to get the subtotal- Apply Discount (if any) to the subtotal- Calculate the Total Price of all items in the cart
Conditions	<ul style="list-style-type: none">- Quantity must be greater than zero- Discount applies only if it meets the criteria (e.g., on sale)

Note: Discount is available on Electronic Items (Smart phones) of 12%.

Items: Smartphones, Refrigerator, Television, Cameras, Projectors, Speakers

National University of Computer and Emerging Science

CLO # 2: Gain hands on experience in writing code that provides the use of logical and bitwise operators to perform efficient data manipulation and apply decision and nested decision structures in control flow to create dynamic, condition-based logic within C-code.

Q2. [7 marks] You are designing a warehouse inventory management system that uses a 6-digit code to manage stock levels and access permissions to restricted items. The system processes each digit in the code to determine the level of access and stock status. Here's how the code is interpreted:

- **Digits 1, 2, and 3** represent item priority and restocking status. These digits are combined into a 3-bit binary number. The resulting binary number determines the stock level status: Each digit represents each item (There are a total of three items) so the binary values for three items are different.
 - **000:** Item is fully stocked.
 - **001:** Item is low in stock, restocking required.
 - **010:** Item is out of stock, urgent restocking needed.
 - **111:** Restricted item, access requires manager approval.
- **Digits 4, 5, and 6** represent security conditions for accessing restricted areas (e.g., locked sections, hazardous materials, or special clearance requirements):
 - **Digit 4:** If the digit is prime, the system restricts access to the hazardous materials area unless the user has special clearance.
 - **Digit 5:** If the digit is even and is a multiple of 4, the system prompts that the item requires temperature-controlled storage.
 - **Digit 6:** If the digit is odd, it indicates the item is imported else its made locally

Example Output:

Enter Number: 214725

- 1 Digit 2 (Starting From MSB): Item 1 is out of stock
- 2 Digit 1: Item 2 is low in stock, restocking required
- 3 Digit 4: Item 3 No change
- 4 Digit 7: Item 1 is Hazardous material
- 5 Digit 2: Item 2 does not require temperature-controlled storage.
- 6 Digit 5: Item 3 Is imported

CLO # 3: Understand and implement code utilize loops for iteration and arrays for data storage and manipulation, demonstrating efficient traversal and management of elements in C-code, while optimizing control flow and enhancing program efficiency.

Q3. [8 marks] Write a program that takes the string "HELLO-WORLD" and reverses it to "DLROW-OLLEH," but with a twist. The string should be converted to one character at a time, and the user will interact with the program as follows:

1. The user will have unlimited attempts to guess the ASCII code for the character that matches the reversed string's position.
2. Once the user correctly guesses the ASCII code for the current character, the program will move on to the next character.
3. If the user's first guess for the next character is incorrect, the program will reset to the original string, and the user will have to start from the beginning.
4. This behavior will continue until the entire string is correctly converted to "DLROW-OLLEH."

Additional Requirements:

- The ASCII values input by the user must be used to convert the characters.
- The reversed words "DLROW" and "OLLEH" should be stored in two separate arrays.
- After successfully reversing the string, the program should display a specific pattern (on the left) using the newly created arrays.

Example Output:

String = HELLO-WORLD

Enter number for the value to change = -4

String =DELLO-WORLD

Enter number for the value to change: = 7

String = DLLLO-WORLD

Enter number for the value to change: = 1

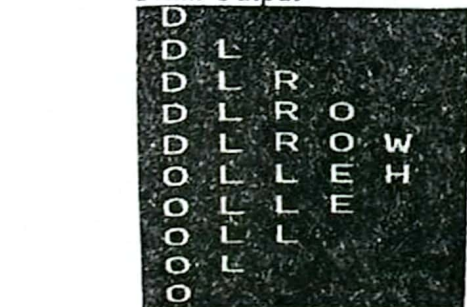
Wrong Guessing Reverting string back

String = HELLO-WORLD

Enter number for the value to change: = -4

(Cycle Continues)

Final Output



Note: You can use multiple (single) loops, but no nesting of loops is allowed for this pattern.