National University of Computer and Emerging Sciences

Programming Fundamentals Lab (CL1002)

Date: 17/10/2024

Course Instructor(s)

Muhammad Monis, Shaheer Ahmad

Lab Mid Exam (A)

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] Conver the following pac chart to C-code

Column	Details
Given Data (Inputs)	**- Employee Salary** **- Contribution Percentage** **- Employer Matching Percentage**
Required Results (Outputs)	**- Total 401(k) Contribution**
Processing	**- Multiply salary by contribution percentage for employee contribution.** **- Calculate employer match and add to employee contribution.**
Conditions	**- Total contribution must not exceed IRS limits.**

Employee Contribution=Employee Salary×(100/Contribution Percentage)

Employer Contribution=Employee Contribution×(100/Employer Match Percentage)

Total Contribution=Employee Contribution+Employer Contribution

Max Contribution: 15%

National University of Computer and Emerging Sciences

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 developing a security system for a bank's vault, which uses an advanced access control mechanism. The system accepts a **6-digit security code** and performs several checks to determine if the user is authorized. The vault has two levels of access: **Administrator** and **User.** **Each of the 6 digits** in the security code represents a certain status of the user, and the system must process these digits (use your mind).

Here's how the system works:

- **Digits 1,2 and 3** of the security code represent **user permissions.** These are combined into a **9-bit binary number** . The resulting binary number defines the type of access: **(Take the first three bits starting from the Most significant Bit)**
    - 000001000: No access (access denied)
    - 001010100: Read-only access (User)
      64 16 4
    - 011111111: Read and Write access (User)
      128
    - 110000001: Full access (Administrator)
      256
- **Digits 4, 5, and 6** of the security code represent **environmental conditions** (e.g., time of day, number of failed attempts, and system status). These conditions are used to determine if access is restricted:
    - If **digit 4** is even, the system restricts access to normal working hours (9 AM to 5 PM).
    - If **digit 5** is odd, the system checks if the number is a multiple of 7 if true the system should print "Access to system resource granted" else "User Access granted" and if the condition is false "Denied Access"
    - If **digit 6** is 1, a hardware failure has been detected, and access is denied.
    - Use Ternary for Digits 4,5 and 6.

Example Output:

Enter Number: 385651
Digit 385 (Starting From MSB): Full Access to the user
~~Digit 6: Access restricted from 9Am to 5Pm~~
~~Digit 4: Item 3 No change~~
~~Digit 7: User Access Granted~~
~~Digit 6: Hardware Failure Detected~~

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 "AC17D68DE" and converts it to "FAST-NUCES," but with a twist. The conversion process should be done one character at a time, with the user interacting with the program as follows:

1. The user will have unlimited attempts to guess the ASCII code for the first character of the target string ("FAST-NUCES").
2. Once the user correctly guesses the ASCII code for the current character, the program will move on to the next character in the sequence.
3. If the user's first guess for the next character is incorrect, the program will reset to the original string, and the user must start from the beginning.
4. This behavior will continue until the entire string is successfully converted to "FAST-NUCES."

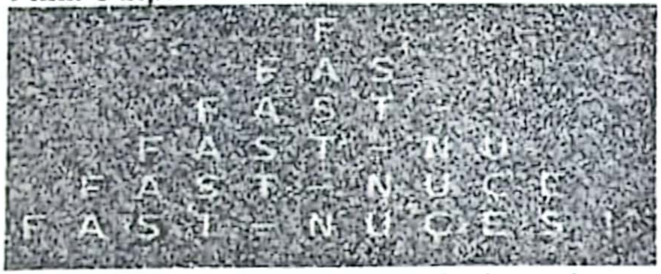
Additional Requirements:

- The ASCII values input by the user must be used to convert the characters.
- After successfully converting the string, a specific pattern should be displayed on the console.

Example Output:

```
String =AC17D68DE
Enter number for the value to change = 5
String =FC17D68DE
Enter number for the value to change: = -2
String =FA17D68DE
Enter number for the value to change: = 1
Wrong Guessing Reverting string back
String =AC17D68DE
Enter number for the value to change: = 5
(Cycle Continues)
```

Final Output:



```
          F
        F A S
      F A S T -
    F A S T - N U
  F A S T - N U C E
F A S T - N U C E S !
```

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