Section 16

PDS Lab **Lab-4** **08.09.2023**

Instructions:

- This lab is based on the topics: Branching and Conditionals.
- You should save each program with the file name as specified against each problem as <Lab#>_<Assignment#>_<Roll#>.c. For example, **04_01_23CS10006.c** to save Program to 1st assignment in Lab 4 with Roll Number 23CS10006
- You should upload each program to the Moodle system. Also, copy+paste your programs to the text window on the test page.
- There will be no evaluation and hence grade, if you don't submit your .c files to the Moodle server. Use **emacs** editor and **gcc command** in terminal to run the following programs.
- Document your programs meaningfully using appropriately named variable and sufficient amount of comments. Documentation and proper code indentation carry marks.
- Do not use advanced concepts like arrays or functions anywhere in your code.
- **The top two lines of your programs must contain the following information:**

```
//Roll No.: <Type in your roll no.>
//Name: <Type in your name>
```

1. Write a program to compute and display the taxi fare for the following situation. The total number of Kilometers travelled and the hours the taxi was retained will be input by the user as two integer numbers. Taxi fare is computed based on the fuel consumption as per the following chart and vehicle rental as described below. If the number of hours retained is less than 1 hour, then there is no rental, for retaining between 1 hour to 12 hours, rental is charged at a flat rate of Rs. 200/-. For retaining between 12 hours and 24 hours, the rental is incremented by Rs. 300/-. For period exceeding a day, additional Rs. 300/- is charged for every additional day or part of a day.

First 8 Km or less	**20 Rs**
Next 4 Km or less	**10 Rs/Km**
Next 4 Km or less	**8 Rs/Km**
Next 4 Km or less	**6 Rs/Km**
Beyond 20 Km	**5 Rs/Km**

2. Write a program to generate 20 random numbers between 1 and 100, display these random numbers. Finally, display the maximum number that was generated.

(Note: 50% penalty will apply for using Arrays in your program)

3. Write a program that takes a positive integer n and checks whether the digits of n form a palindrome or not.
(Note: A number is said to be a palindrome if it is equal to itself in the reverse order. Thus 16461 is a palindrome whereas 12212 is not.)
(Hint: To be able to check for palindrome, you first need to generate the reverse of the number. That is for 1235, you need to generate a number that is 5321. Then compare with the original number.)

4. Write the following program. In the main program, in an infinite loop keep on prompting the user “enter a single digit number” and read the number. As soon as the user enters a 1 that was immediately preceded by entry of 5 and was succeeded by entry of 7 after entry of any one number, then your program should display “pattern found”. The program should terminate when a negative number is entered.

Example: User enters **3 2 5 5 2 5 1 9 7**. The program should display “pattern found” after number 7 is entered.

5. Write a program that will read an integer value from the user and display the sum of the digits of the number and also display the largest and second largest digit in the number. If the entered integer value has only one digit, then it should display 0 for the second largest digit.

Example: User enters 3 2 5 5
The program should display:
Sum of digits=15
Largest = 5 Second Largest =3

6. Generate 20 random integers between 5000 and 6000. Display the numbers and also the sum of the digits for each number. Display the number having smallest sum of digits**.**

(Penalty will apply if you use arrays in your program)