BTECH CSE SEM 2 DIV 1 C PROGRAMMING PRACTICAL LIST PHASE-1

1. Implement a program in GNU C to display the message “Hello World!” on screen.
2. Implement a program in GNU C where the user enters an integer from keyboard at run-time. The entered integer should be displayed on screen.
3. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The sum of both the integers should be displayed on screen.
4. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The sum, difference and product of both the integers should be displayed on screen.
5. Implement a program in GNU C where the user enters a floating point number (decimal number) from keyboard at run-time. The entered number should be displayed on screen.
6. Implement a program in GNU C where the user enters two floating point numbers from keyboard at run-time. The sum, difference and product of both the numbers should be displayed on screen.
7. Implement a program in GNU C where the user enters a character from keyboard at run-time. The entered character and its corresponding ASCII value (in decimal) should be displayed on screen.
8. Implement a program in GNU C where the user enters the radius of a circle from keyboard at run-time. The circumference and area of the circle should be displayed on screen. DO NOT USE any library function of math.h for this program.
9. Implement a program in GNU C where the user enters the length of a side of a square from keyboard at run-time. The perimeter and area of the circle should be displayed on screen. Make use of pow() library function of math.h for this program.
10. Implement a program in GNU C where the user enters the temperature in celcius unit from keyboard at run-time. The corresponding temperature in Fahrenheit unit should be displayed on screen.
11. Implement a program in GNU C where the user enters the temperature in Fahrenheit unit from keyboard at run-time. The corresponding temperature in Celcius unit should be displayed on screen.
12. Implement a program in GNU C where the user enters the weight in kilograms unit from keyboard at run-time. The corresponding weight in grams unit should be displayed on screen.
13. Implement a program in GNU C where the user enters the weight in grams unit from keyboard at run-time. The corresponding weight in kilograms unit should be displayed on screen.
14. Implement a program in GNU C to swap the values of two variables. The value of both the variables before swapping and after swapping should be displayed on screen.
15. Implement a program in GNU C where the user enters two integers a and b from keyboard at run-time. Accordingly, the message “a is greater than b”, “a is less than b” or “a is equal to b” should be displayed on screen.
16. Implement a program in GNU C where the user enters an integer from keyboard at run-time. Accordingly, the message “ The entered integer is odd” or “The entered integer is even” should get displayed on screen.
17. Implement a program in GNU C where the user enters the percentage result obtained from keyboard at run-time. The corresponding grade should be displayed on screen. Assume that the rules of grading are as follows: for % <40, grade is FAIL, for %>=40 and <=49, grade is

PASS, for % >=50 and <=59 grade is SECOND, for % >=60 and <=69 grade is FIRST, for % >=70 grade is DISTINCTION. Make use of if /if-else/else constructs for above program.

18. Implement above program using SWITCH-CASE construct.
19. Implement a program in GNU C where the user enters three integers from keyboard at run-time. The greatest of all the three numbers should be displayed on screen. DO NOT USE ARRAYS.
20. Implement a program in GNU C where the user enters an alphabetic character from keyboard at run-time. The program should classify the entered alphabet as “VOWEL” or “CONSONANT”.
21. Implement a program in GNU C where the user enters a character from keyboard at run-time. The program should classify whether the entered character is an UPPER CASE alphabet, LOWER CASE alphabet, numeric or OTHER character. DO NOT USE ANY LIBRARY FUNCTIONS except printf and scanf.
22. Implement a program in GNU C where the user enters a multi-digit integer from keyboard at run-time. The program should extract the last digit of the number and display its value.
23. Implement a program in GNU C where the user enters a multi-digit integer from keyboard at run-time. The program should display the total number of digits in that number. Eg. If 23 is entered, the output of the program should be 2. Assume that no leading zeroes are entered while entering the integer.
24. Implement a program in GNU C to print the message “Hello World” 20 times, without replicating the code.
25. Implement a program in GNU C where the user enters a natural number from keyboard at run-time. The program should display the SIGMA or sum of integers starting from 1 up to the given number.
26. Implement a program in GNU C where the user enters a natural number from keyboard at run-time. The program should display the factorial of the given number.
27. Implement a program in GNU C where the user enters a natural number from keyboard at run-time. The program should classify the number as a prime, composite or a special number.
28. Implement a program in GNU C where the user enters a multi-digit integer from keyboard at run-time. The program should display the sum of digits in that number. Eg. If 23 is entered, the output of the program should be 5.
29. Implement a program in GNU C where the user enters a multi-digit integer from keyboard at run-time. The program should display the reversed number. Eg. If 23 is entered, the output of the program should be 32. For single digit number, the output should remain same as the original number entered.
30. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The program should display the GCD of the two integers.
31. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The program should display the LCM of the two integers.
32. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The program should display whether the first integer is exactly divisible by the second integer.
33. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The program should display whether the first integer is a factor of the second integer.

34. Implement a program in GNU C where the user enters two integers from keyboard at run-time. The program should display whether the first integer is a multiple of the second integer.
35. Implement a program in GNU C where the user enters a decimal integer from keyboard at run-time. The program should display the equivalent binary, octal and hexadecimal value of the entered number.
36. Implement a program in GNU C where the user enters a binary integer from keyboard at run-time. The program should display the equivalent decimal, octal and hexadecimal value of the entered number.
37. Implement a program in GNU C where the user enters a hexadecimal integer from keyboard at run-time. The program should display the equivalent binary, octal and decimal value of the entered number.
38. Implement a program where the user is supposed to enter an even integer. Till the user doesn't enter an even integer, the program continuously prompts the user that the entered number is incorrect and to kindly enter an even integer.
39. Implement a program where the user is supposed to enter an even integer. Till the user doesn't enter an even integer, the program continuously prompts the user that the entered number is incorrect and to kindly enter an even integer upto 4 times. After 4^{th} consecutive failure to enter the correct integer, the program terminates.
40. Implement a program where the user is supposed to enter 5 odd integers. The program should display the sum of all the entered odd integers. In case, the user enters an even integer, then the entered even integer SHOULD NOT BE COUNTED while calculating the sum.
41. Implement a program where the user is supposed to enter 5 odd integers. The program should display the sum of all the entered odd integers. In case, the user enters an even integer, then the program displays the sum of the entered odd integers up to the last odd integer entered and terminates.
42. Implement a program which displays the following menu when the program is run.
    1-ACCOUNT OPENING
    2-DEPOSIT
    3-WITHDRAW
    4-BALANCE ENQUIRY
    5-ACCOUNT CLOSING
    6-ACCOUNT DEACTIVATE
    Depending upon the number entered by the user, the corresponding message will be displayed. Eg. If the user enters 4, the message "BALANCE ENQUIRY" will be displayed.