# Bankura Unnayani Institute of Engineering

Practical Assignment – odd – 2021-22
COMPUTER SCIENCE AND ENGINEERING
&
INFORMATION TECHNOLOGY
IT Workshop PYTHON Lab
Paper Code:- PCC-CS393

1. **Exercise 1: Celsius to Fahrenheit and Kelvin**
   Write a program that begins by reading a temperature from the user in degrees Celsius. Then your program should display the equivalent temperature in degrees Fahrenheit and degrees Kelvin. The calculations needed to convert between different units of temperature can be found on the Internet.
2. **Exercise 2: Area of a Room**
   Write a program that asks the user to enter the width and length of a room. Once these values have been read, your program should compute and display the area of the room. The length and the width will be entered as floating-point numbers. Include units in your prompt and output message; either feet or meters, depending on which unit you are more comfortable working with.
3. **Exercise 3: Area of a Field**
   Create a program that reads the length and width of a farmer's field from the user in feet. Display the area of the field in acres.
   Hint: There are 43,560 square feet in an acre.
4. **Exercise 4: Compound Interest**
   Pretend that you have just opened a new savings account that earns 4 percent interest per year. The interest that you earn is paid at the end of the year, and is added to the balance of the savings account. Write a program that begins by reading the amount of money deposited into the account from the user. Then your program should compute and display the amount in the savings account after 1, 2, and 3 years. Display each amount so that it is rounded to 2 decimal places.
5. **Exercise 5: Arithmetic**
   Create a program that reads two integers, a and b, from the user. Your program should compute and display:
   • The sum of a and b
   • The difference when b is subtracted from a
   • The product of a and b
   • The quotient when a is divided by b
   • The remainder when a is divided by b
   • The result of log10 a
   • The result of a
   Hint: You will probably find the log10 function in the math module helpful for computing the second last item in the list.
6. **Exercise 6: Sum of the Digits in an Integer**

Develop a program that reads a four-digit integer from the user and displays the sum of its digits. For example, if the user enters 3141 then your program should display 3+1+4+1=9.

7. **Exercise 7: Even or Odd?**
   Write a program that reads an integer from the user. Then your program should display a message indicating whether the integer is even or odd.
8. **Exercise 8:Vowel or Consonant**
   In this exercise you will create a program that reads a letter of the alphabet from the user. If the user enters a, e, i, o or u then your program should display a message indicating that the entered letter is a vowel. If the user enters y then your program should display a message indicating that sometimes y is a vowel, and sometimes y is a consonant. Otherwise your program should display a message indicating that the letter is a consonant.
9. **Exercise 9:Leap Year**
   Show that a month is a leap year or not.
10. **Exercise 10: Discount**
    While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If quantity and price per item are input through the keyboard, write a program to calculate the total expenses.
11. **Exercise 11: Employee Salary**
    In a company an employee is paid as under: If his basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% of basic salary. If his salary is either equal to or above Rs. 1500, then HRA = Rs. 500 and DA = 98% of basic salary. If the employee's salary is input through the keyboard, write a program to find his gross salary.
12. **Exercise 12: Percentage of marks**
    Percentage marks obtained by a student are input through the keyboard. The student gets a division as per the following rules:
    Percentage above or equal to 60 - First division
    Percentage between 50 and 59 - Second division
    Percentage between 40 and 49 - Third division
    Percentage less than 40 - Fail
    Write a program to calculate the division obtained by the student.

13. **Exercise 13: Points on a straight line**
    Given three points (x1, y1), (x2, y2) and (x3, y3), write a program to check if all the three points fall on one straight line.

14. **Exercise 14: (Find future dates)**
    Write a program that prompts the user to enter an integer for today’s day of the week (Sunday is 0, Monday is 1, ..., and Saturday is 6). Also prompt the user to enter the number of days after today for a future day and display the future day of the week. Here is a sample run:

Enter today's day:1
Enter the number of days elapsed since today:3
Today is Monday and the future day is Thursday

Enter today's day:0
Enter the number of days elapsed since today:31
Today is Sunday and the future day is Wednesday

15. **Exercise 15: Average**
    Write a program that finds the average of first n numbers using a for loop.
16. **Exercise 16: Series**
    Write a program to find sum of series $\frac{1^2}{1} + \frac{2^2}{2} + \frac{3^3}{3} \ldots + \frac{n^n}{n}$
17. **Exercise 17: Series**
    Write a program that prompts users to enter numbers. Once a user enters -1, it displays the count, sum, and average of even numbers and that of odd numbers.
18. **Exercise 18: Prime Numbers**
    Write a Python program to print the prime numbers for a user provided range.
19. **Exercise 19: Not divisible by 2 and 3**
    Write a program that displays all the numbers from 1-100 that are not divisible by 2 as well as by 3.
20. **Exercise 20: Sum of odd numbers**
    Write a program to print the sum of all odd numbers from 1 to 100.
21. **Exercise 21: Binary representation**
    Write a program that reads an integer. If it is positive then display its binary equivalent. If it is negative then ignore it and ask to re-enter different numbers. Program must stop when the user input 999.
22. **Exercise 22: Pattern**
    Write a program to generate following pattern:

23. **Exercise 23: Pattern**
    Write a program to generate following pattern:

24. **Exercise 24: Pattern**

    Write a program to generate following pattern:

25. **Exercise 25: Sequence**

    Write a program to generate following sequence:
    1, 8, 27, 64,.....

26. **Exercise 26: Sequence**

    Write a program to generate following sequence:
    -5, -2, 0, 3, 6, 9, 12,.....

27. **Exercise 27: Sequence**

    Write a program to generate following sequence:
    -2, -4, -6, -8, -12,.....

28. **Exercise 27: Sequence**

    Write a program to generate following sequence:
    1, 4, 7, 10,.....

29. **Exercise 29: largest and smallest**

    Write a program that reads integers until the user wants to stop. When the user stops entering numbers, display the largest and smallest number entered.

30. **Exercise 30: Sum of the series**

    Write a program to print the sum of the following $n^{th}$ series:

$-x+x^2-x^3+x^4+$.....

**31. Exercise 31: Sum of the series**

Write a program to print the sum of the following $n^{th}$ series:

$1+(1+2)+(1+2+3)+$.....

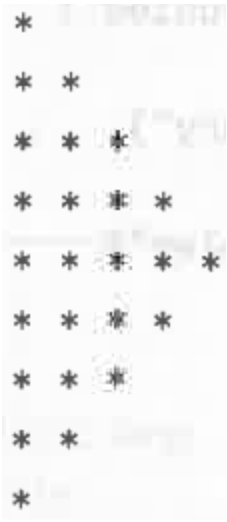
**32. Exercise 32: Sum of the series**

Write a program to print the sum of the following $n^{th}$ series:

$1-x+\frac{x^2}{2!}-\frac{x^3}{3!}+$.....

**33. Exercise 33: Pattern**

Write a program to generate following pattern:

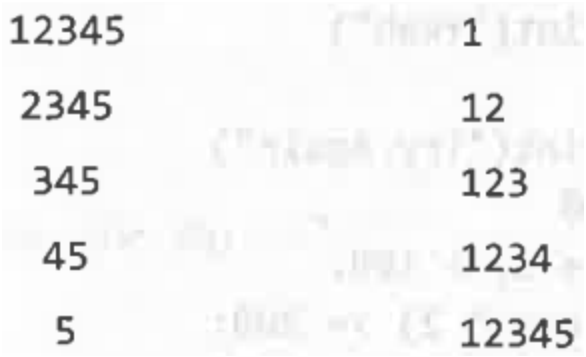


**34. Exercise 34: Pattern**

Write a program to generate following pattern:


**35. Exercise 35: Pattern**

**Write a program to read a 5 digit number and then display the number in the following format. For example if user entered 12345, the result should be:**




36. **Exercise 36:LIST**
    Create a list of 10 integers, Write a program to search an element from the list using binary search.
37. **Exercise 37:LIST**
    Create a list of 10 integers, Write a program to sort elements of the list in ascending order using bubble sort.

38. **Exercise 38:LIST**
    Write a program that will create a list of numbers from 1-20. Then delete all odd numbers from the list.
39. **Exercise 39:LIST**
    Write a program to create an empty list. Then ask the user to input an integer number, if a positive number is entered then add it to the list otherwise ignore the input. Finally find the sum of all elements in the list.
40. **Exercise 40:LIST**
    Create two lists of the same length and of integer type, Find the sum of each element of both the lists. Finally store resultant elements in a third list.
41. **Exercise 41:LIST**
    Create a 2D list of size 3X5. Then find and print the average of each row of the list.
42. **Exercise 42:Dictionary**
    Create a dictionary that will contain 5 products with their price. Then find and print the most valuable item with its price.
43. **Exercise 43:Dictionary**
    Create an empty dictionary named passed. Then ask the user to enter the student name and percentage of marks until the user wants to stop. If the student achieves 60% and more then only it will be added to the dictionary. Finally print the whole dictionary.
44. **Exercise 44:Function**
    Write a program that finds the sum of two integers using a function.
45. **Exercise 45:Function**
    Write a program that finds the greatest of three numbers using function. Pass the numbers as arguments.
46. **Exercise 46:Function**

A prime number is an integer greater than one that is only divisible by one and itself. Write a function that determines whether or not its parameter is prime, returning True if it is, and False otherwise. Write a main program that reads an integer from the user and displays a message indicating whether or not it is prime.

47. **Exercise 47:Function**

    Create a function named nextPrime that finds and returns the first prime number larger than some integer, n. The value of n will be passed to the function as its only parameter. Include a main program that reads an integer from the user and displays the first prime number larger than the entered value.

48. **Exercise 48:Twin primes**

    Twin primes are a pair of prime numbers that differ by 2. For example, 3 and 5, 5 and 7, and 11 and 13 are twin primes. Write a program to find all twin primes less than 1,000. Display the output as follows:
    (3, 5)
    (5, 7)...

49. **Exercise 49: Is Palindrome**

    A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and call a function isPalindrome() to report whether the integer is a palindrome.

50. **Exercise 50:**

    Write a function that accepts three integers, and returns True if they are sorted, otherwise it returns False.