**PSG College of Technology, Coimbatore-04**

**Department of Applied mathematics and Computational Sciences**

**20XC28 – Python Programming Lab**

**Problem Sheet – III**

1. Write a program that prompts the user to enter a number within the range of 1 through 10. The program should display the Roman numeral version of that number. If the number is outside the range of 1 through 10, the program should display an error message. The following table shows the Roman numerals for the numbers 1 through 10.



1. Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

1. Electromagnetic radiation can be classified into one of 7 categories according to its frequency, as shown in the table below:



Write a program that reads the frequency of the radiation from the user and displays the appropriate name.

1. Write a program that reads in three integers, a, b, and c, representing the lengths of the sides of a triangle and say whether the resulting triangle is isosceles or equilateral or neither. An isosceles triangle is a triangle with (at least) two equal sides. An equilateral triangle is a triangle with three equal sides. Test your program with the following values of a, b and c. (Use Iteration to take any number of inputs)

Input 1: 3, 4, 5 Output: Given Triangle is not an isosceles or an equilateral

Input 2: 5, 5, 2 Output: Given Triangle is an isosceles

Input 3: 10, 6, 6 Output: Given Triangle is an isosceles

Input 4: 7, 7, 7 Output: Given Triangle is an equilateral

1. The monthly payment for a given loan pays the principal and the interest. The monthly interest is computed by multiplying the monthly interest rate and the balance (the remaining principal).

The principal paid for the month is therefore the monthly payment minus the monthly interest. Write a program that lets the user enter the loan amount, number of years, and interest rate, and then displays the amortization schedule for the loan. Here is a sample run:



1. Write a program that displays a multiplication table that shows the products of all combinations of integers from 1 times 1 up to and including 10 times 10. Your multiplication table should include a row of labels across the top of it containing the numbers 1 through 10. It should also include labels down the left side consisting of the numbers 1 through 10. The expected output from the program is shown below:



1. The prime factorization of an integer, *n*, can be determined using the following steps.



Write a program that reads an integer from the user. If the value entered by the user is less than 2 then your program should display an appropriate error message. Otherwise your program should display the prime numbers that can be multiplied together to compute n, with one factor appearing on each line. For example:



1. One of the first known examples of encryption was used by Julius Caesar. Caesar needed to provide written instructions to his generals, but he didn’t want his enemies

to learn his plans if the message slipped into their hands. As result, he developed what later became known as the Caesar Cipher.

The idea behind this cipher is simple (and as a result, it provides no protection against modern code breaking techniques). Each letter in the original message is shifted by 3 places. As a result, A becomes D, B becomes E, C becomes F, D becomes G, etc. The last three letters in the alphabet are wrapped around to the beginning: X becomes A, Y becomes B and Z becomes C. Non-letter characters are not modified by the cipher.

Write a program that implements a Caesar cipher. Allow the user to supply the message and the shift amount, and then display the shifted message. Ensure that your program encodes both uppercase and lowercase letters. Your program should also support negative shift values so that it can be used both to encode messages and decode messages.

1. What’s the minimum number of times you have to flip a coin before you can have three consecutive flips that result in the same outcome (either all three are heads or all three are tails)? What’s the maximum number of flips that might be needed? How many flips are needed on average? In this exercise we will explore these questions by creating a program that simulates several series of coin flips.

Create a program that uses Python’s random number generator to simulate flipping a coin several times. The simulated coin should be fair, meaning that the probability of heads is equal to the probability of tails. Your program should flip simulated coins until either 3 consecutive heads of 3 consecutive tails occur. Display an H each time the outcome is heads, and a T each time the outcome is tails, with all of the outcomes shown on the same line. Then display the number of flips needed to reach 3 consecutive flips with the same outcome. When your program is run it should perform the simulation 10 times and report the average number of flips needed.

Sample output is shown below:



1. A baseball player’s batting average is calculated as the number of hits divided by the official number of at-bats. In calculating official at-bats, walks, sacrifices, and occasions when hit by the pitch are not counted. Write a program that takes an input containing total number of players, and for each player player number and batting records. Trips to the plate are coded in the batting record as follows: H—hit, O—out, W—walk, S—sacrifice, P—hit by pitch. The program should output for each player the input data followed by the batting average. (Note: Number of batting records may vary from player to player. So for each batting record get option to read next record.

Sample input:

Total Number of players : 3

Enter Player Number : 12

Enter batting record : H

You wanna continue? [y/n] y

Enter batting record : H

You wanna continue? [y/n] y

Continues …..

So, for example

12 HOOOWSHHOOHPWWHO

4 OSOHHHWWOHOHOOO

7 WPOHOOHWOHHOWOO

Corresponding output:

Player 12's record: HOOOWSHHOOHPWWHO

Player 12's batting average: 0.455

Player 4's record: OSOHHHWWOHOHOOO

Player 4's batting average: 0.417

Player 7's record: WPOHOOHWOHHOWOO

Player 7's batting average: 0.364

1. Credit card numbers follow certain patterns. A credit card number must have between 13 and 16 digits. The number must start with the following:

* 4 for Visa cards
* 5 for MasterCard cards
* 37 for American Express cards
* 6 for Discover cards

In 1954, Hans Luhn of IBM proposed an algorithm for validating credit card numbers. The algorithm is useful to determine whether a card number is entered correctly or is scanned correctly by a scanner. Almost all credit card numbers are generated following this validity check, commonly known as the Luhn check or the Mod 10 check. It can be described as follows. (For illustration, consider the card number 4388576018402626).

1. Double every second digit from right to left. If doubling of a digit results in a two-digit number, add the two digits to get a single digit number.



1. Now add all single-digit numbers from Step 1.



1. Add all digits in the odd places from right to left in the card number.



1. Sum the results from Step 2 and Step 3.



1. If the result from Step 4 is divisible by 10, the card number is valid; otherwise, it is invalid. For example, the number 4388576018402626 is invalid, but the number 4388576018410707 is valid.

Write a program that prompts the user to enter a credit card number and display whether the number is valid or not.

1. Suppose you save $100 each month into 0.05/12 = 0.00417 a savings account with the annual interest rate 5%. So, the monthly interest rate is After the first month, the value in the account becomes



and so on.

Write a program that prompts the user to enter an amount (e.g., 100), the annual interest rate (e.g., 5), and the number of months (e.g., 6), and displays the amount in the savings account after the given month.

1. Write a program that asks the user to enter today’s sales for five stores. The program should then display a bar graph comparing each store’s sales. Create each bar in the bar graph by displaying a row of asterisks. Each asterisk should represent Rs.100 of sales.

Here is an example of the program’s output.

Enter today's sales for store 1: 1000

Enter today's sales for store 2: 1200

Enter today's sales for store 3: 1800

Enter today's sales for store 4: 800

Enter today's sales for store 5: 1900

SALES BAR CHART

(Each \* = Rs.100)

Store 1: \*\*\*\*\*\*\*\*\*\*

Store 2: \*\*\*\*\*\*\*\*\*\*\*\*

Store 3: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Store 4: \*\*\*\*\*\*\*\*

Store 5: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Also, your program should print the average sale.

1. SMITH NUMBER is a composite integer with the property that the sum of its digits is the same as the sum of the digits of its prime factors. A composite integer N whose digit sum S(N) is equal to the sum of the digits of its prime factors Sp (N) is called a Smith number. For example: 85 is a Smith number because digit sum of 85 i.e. S(85) = 8 + 5=13, which is equal to the sum of the digits of its prime factors. i.e.

Sp (85) = Sp (5 x 17) = 5 + 1 + 7 = 13.

Test Data: 22, 58, 85, 94, 166, 202, 265, 274

Write a program to read a composite integer and check whether it is a smith number

or not.

1. Use nested loops that display the following patterns in four separate programs:



1. Write a program to print the calendar for the given month. Input data are (1) day of the week of an input month, and (2) number of days of the input month. When input figure is not appropriate, e.g. day of the week is not within 0 to 6, or month is not within 1 to 12, an error message should be shown.

**Sample**

Which day of the week does the month start?

Input figure among following figures.

0:Sun, 1:Mon, 2:Tue, 3:Wed, 4:Thu, 5:Fri, 6:Sat

**5**

How many days does the month have?

**28**

