

MCT-242 : COMPUTER PROGRAMMING-I

using Python 3.9



Prepared By:
Mr. Muhammad Ahsan Naeem



YouTube Playlist

https://www.youtube.com/playlist?list=PLWF9TXck7O_wMDB-VriREZ6EvwkWLNb7q

LAB 14: PRACTICE SESSION

Tasks:

- [1] **Survey for favorite Beverages:** Write a program that performs a survey tally on beverages. The program will ask user to enter favorite beverage of first person and then should prompt for the next person until a value 0 is entered to terminate the program. Each person participating in the survey should choose their favorite beverage from the following list:

1. Coffee 2. Tea 3.Coke 4. Orange Juice

After the survey is complete indicated by entering a 0, the program should display the result in form of total number of persons participating in the survey and number of votes against each beverage.

Sample output is:

```
*****
1. Coffee      2. Tea      3.Coke      4. Orange Juice
*****
This is survey for the favorite Beverage.

Choose (1-4) from the above menu or 0 to exit the program.

Please input the favorite beverage of person #1: 4
Please input the favorite beverage of person #2: 1
Please input the favorite beverage of person #3: 3
Please input the favorite beverage of person #4: 1
Please input the favorite beverage of person #5: 1
Please input the favorite beverage of person #6: 0

The results are as follows:
The total number of people participated: 5

Beverage      Number of Votes
*****
Coffee        3
Tea           0
Coke          1
Orange Juice  1
```

- [2] An integer is called **squarefree** if it is not divisible by any perfect squares other than 1. For instance, 42 is squarefree because its divisors are 1, 2, 3, 6, 7, 21, and 42, and none of those numbers (except 1) is a perfect square. On the other hand, 45 is not squarefree because it is divisible by 9, which is a perfect square. Write a program that asks the user for an integer and tells them if it is squarefree or not. You have to do this task by creating two functions:

- A function `isPerfSq()` with Boolean output that will decide if the input number is perfect square or not.
- A function `isSqFree()` with Boolean output which will use `isPerSq()` to decide if the input number is square free or not.

The structure of the complete program will look like:

```
def isPerfSq(x):
    'Finds if x is perfect square or not'
    #Complete the logic
def isSqFree(x):
    'Finds if x is square free or not'
    #Complete the logic

### Main Program starts from here ###
num=eval(input('Enter a number: '))
if(isSqFree(num)):
    print(f'{num} is a Square Free number!')
else:
    print(f'{num}is Not a Square Free number!')
```

[3] A number is known as to be a **palindrome** if it same in reverse order e.g. 1331, 4536354, 787 are palindromes and 1330 and 1778 are not. Write a program that will take a number from user as input and will display if the number is palindrome or not.

Sample output is:

```
Enter a number: 1331
Palindrome number
```

Another Sample output is:

```
Enter a number: 1330
Not a Palindrome number
```

How to do it:

The core part of the program is to generate a number in reverse order as that of the input number. Once it is generated, we need simple comparison of reversed number and the input number.

To generate the reversed number, we will use similar logic as used in last task of the last lab session. There, we did repeated task of picking the unit digit and getting the remaining number without unit digit. Within the same logic we will generate the reversed number from picked unit digit. This is better explained with an example number as `x=5429`.

x	a=x%10	x=x//10	rev=?
5429	9	542	9
542	2	54	92
54	4	5	924
5	5	0	9245

The iterations are controlled with the same condition as done in previous task. You need to add one more statement to calculate reverse number. From above table, you can find the expression for reverse number in last column.

[4] Write a program that will print all palindrome numbers from 1 to 10000. Use a **for** loop to iterate from 1 to 10000 and a **while** loop to check if it is palindrome. Don't create any function for this task.

[5] Repeat task 6 of previous lab session with added features as described below:

- After displaying number of zeros, the program should ask if user wants to enter another one and proceed accordingly.

Sample Output is:

```
Enter a number: 5402
No. of zeros in 5402= 1
Press Y if you enter another number: y

Enter a number: 400
No. of zeros in 400= 2
Press Y if you enter another number: n

Thanks!
```

- Now, instead of displaying number of zeros, ask user for the digit he want to count.

Sample Output is:

```
Enter a number: 5402
Enter single digit: 4
No. of 4 in 5402=1
Press Y if you enter another number: y

Enter a number: 4232
Enter single digit: 2
No. of 2 in 4232=2
Press Y if you enter another number: n

Thanks!
```

- c. Now add a feature in your program that will validate if user is entering a single digit asked by the program. If it is not a single digit, program should ask again until it is a positive single digit.

Sample Output is:

```
Enter a number: 5402
Enter single digit: 4
No. of 4 in 5402=1
Press Y if you enter another number: y

Enter a number: 4232
Enter single digit: 22
Enter single digit: 41
Enter single digit: -2
Enter single digit: 54
Enter single digit: 2
No. of 2 in 4232=2
Press Y if you enter another number: y

Enter a number: 213
Enter single digit: 3
No. of 3 in 213=1
Press Y if you enter another number: n

Thanks!
```

- d. For the single digit entry from user, allow him 3 turns only. If user enters the digit wrongly third time, just exit the program with appropriate message:

Sample Output is:

```
Enter a number: 5402
Enter single digit: 42
Enter single digit: -2
Enter single digit: 4
No. of 4 in 5402=1
Press Y if you enter another number: y

Enter a number: 4232
Enter single digit: 22
Enter single digit: 41
Enter single digit: -2
It seems you don't want to follow the instructions.
Good Bye!
```