Prac_1a
Create a program that asks the user to enter their name and their age. Print out a
message addressed to them that tells them the year that they will turn 100 years old.
CODE:

```
import datetime
name = input("Enter your name: ")
age = int(input("Enter your age: "))
currentyear = datetime.datetime.now().year
dob = currentyear - age
print(dob+100)
```

OUTPUT:

```
Enter your name: Jermin
Enter your age: 22
2102
```

Prac_1b
Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
CODE:

```
num = int(input("Enter a number: "))
if(num%2 == 0):
    print(num, " is Even ")
else:
    print(num, " is Odd ")
```

OUTPUT:

```
Enter a number: 10
10  is Even

Enter a number: 11
11  is Odd
```

Prac_1c
Write a program to generate the Fibonacci series.
CODE:

```
n = int(input("Enter a number: "))
first = 0
second = 1
for i in range(n):
    print(first)
    temp = first
    first = second
    second = second + temp
```

OUTPUT:

```
Enter a number: 8
0
1
1
2
3
5
8
13
```

Prac_1d
Write a function that reverses the user defined value

CODE:

```
#Using function
def rev(n):
    reverse=0
    while(n>0):
        reminder = n % 10
        reverse = (reverse * 10) + reminder
        n = n // 10
    return reverse
n = int(input("Enter a number: "))
reverse1 = rev(n)
print("Reverse number: ", reverse1)
```

OUTPUT:

```
Enter a number: 12345
Reverse number:  54321
```

Prac_1e

Write a function to check the input value is Armstrong and also write the function for Palindrome.

CODE:

```
def Armstrong(n):
    temp = n
    result = 0
    while(temp>0):
        remainder = temp%10
        result = remainder **3 + result
        temp = temp // 10
    if(result==n):
        print(n, "is an Armstrong")
    else:
        print(n, "is not an Armstrong")

def Palindrome(n):
        temp = n
        reverse = 0
        while(temp>0):
            remainder = temp%10
            reverse = reverse * 10 + remainder
            temp = temp // 10
        if(n==reverse):
            print(reverse, "is a Palindrome")
        else:
            print(reverse, "is not a Palindrome")

n = int(input("Enter a number: "))
Armstrong(n)
Palindrome(n)
```

OUTPUT:

```
Enter a number: 153
153 is an Armstrong
351 is not a Palindrome
```

```
Enter a number: 12321
12321 is not an Armstrong
12321 is a Palindrome
```

Prac_1f

Write a recursive function to print the factorial for a given number.

CODE:

```
def fact(n):
    if(n==1):
        return 1
    else:
        return n*fact(n-1)
n = int(input("Enter a number: "))
result = fact(n)
print(result)
```

OUTPUT:

```
Enter a number: 5
120
```

Prac_2a

Write a function that takes a character (i.e. a string of length 1) and returns True
if it is a vowel, False otherwise.

CODE:

```
def vowel(s):
    if(s=='a' or s=='e' or s=='i' or s=='o' or s=='u' or s=='A' or s=='E' or
s=='I' or s=='O' or s=='U'):
        return True
    else:
        return False
s = input("Enter a Character: ")
vowel(s)
result= vowel(s)
print(result)
```

OUTPUT:

```
Enter a Character: a
True
_____________________
Enter a Character: U
True
_____________________
Enter a Character: J
False
```

Prac_2b

Define a function that computes the length of a given list or string

CODE1: (STRING)

```
def findlen(str):
    counter = 0
    for i in str:
        counter = counter + 1
    return counter
str = input("Enter a string: ")
print(findlen(str))
```

OUTPUT FOR STRING:

```
Enter a string: Python language
15
```

CODE2: (LIST)

```
def findlen(lst):
    counter = 0
    for i in lst:
        counter = counter + 1
```

```
    return counter
lst = ["Jermin", 12.5, 56, 67]
print(findlen(lst))

OUTPUT FOR LIST:
4
```

Prac_2c

Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:

```
****
*********
*******
```

CODE:

```
def histogram(lst):
    for i in lst:
        print(i * '*')
lst=[]
ln = int(input("Enter the list length: "))
print("Enter integer ", ln)
for i in range(ln):
    data = int(input())
    lst.append(data)

histogram(lst)

OUTPUT:
Enter the list length: 3
Enter integer  3
1
2
3
*
**
***
```

Prac_3a

A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not

CODE:

```
def pangram(str, alphabet):
    flag = True
    for char in alphabet:
        if char not in str.lower():
            flag = False
    if(flag == True):
        print("Pangram")
    else:
        print("Not a Pangram")
str = "The quick brown fox jumps over the lazy dog"
alphabet = "abcdefghijklmnopqrstuvwxyz"
pangram(str, alphabet)

OUTPUT:
Pangram
```

Prac_3b

Take a list, say for example this one:
 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]and write a program
 that prints out all the elements of the list that are less than 5
CODE:

```
#Method 1
a = [1,1,2,3,5,8,13,21,34,55,89]
for i in a:
    if i<5:
        print(i)


#Method 2
a = [1,1,2,3,5,8,13,21,34,55,89]
lst=[]
for i in a:
    if i<5:
        lst.append(i)
print(lst)
```

OUTPUT:

```
1
1
2
3
[1, 1, 2, 3]
```

Prac_4a
Write a program that takes two lists and returns True if they have at least one common member
CODE:

```
lst1 = [1,2,3,4,5]
lst2 = [5,6,7,8,9]
for x in lst1:
    for y in lst2:
        if(x==y):
            print("True")
```

OUTPUT:

```
True
```

Prac_4b
Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements
CODE:

```
lst = [0,1,2,3,4,5,6,7,8]
print("Original List :", lst)

for i in lst:
    if i not in(0,2,4,5):
        print(i,end=" ")
```

OUTPUT:

```
Original List : [0, 1, 2, 3, 4, 5, 6, 7, 8]
1 3 6 7 8
```

Prac_4c
Write a Python program to clone or copy a list
CODE:

```
original_list1 = [1,2,3,4,5]
print("Original List1: ",original_list1)
new_list=list(original_list1)
print("New List: ",new_list)



#Method 2
original_list2 = [6,7,8,9,10]
print("Original List2 = ",original_list2)
copy=original_list2.copy()
print("COPY :",copy)

OUTPUT:
Original List1:  [1, 2, 3, 4, 5]
New List:  [1, 2, 3, 4, 5]

Original List2 =  [6, 7, 8, 9, 10]
COPY : [6, 7, 8, 9, 10]

--------------------------------------------------------------------------------
------------
Prac_5a
Write a Python script to sort (ascending and descending) a dictionary by value
CODE:
import operator
d = {'C':90, 'C++':100, 'Python':80, 'Java':60}
print("Original dictionary = ",d)

asc = dict(sorted(d.items(), key=operator.itemgetter(1)))
print("Ascending =",asc)

desc = dict(sorted(d.items(), key=operator.itemgetter(1), reverse=True))
print("Descending =",desc)

OUTPUT:
Original dictionary =  {'C': 90, 'C++': 100, 'Python': 80, 'Java': 60}
Ascending = {'Java': 60, 'Python': 80, 'C': 90, 'C++': 100}
Descending = {'C++': 100, 'C': 90, 'Python': 80, 'Java': 60}

--------------------------------------------------------------------------------
-----------
Prac_5b
Write a Python script to concatenate following dictionaries to create a new one.
Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
CODE:
dict1 = {1:10, 2:20}
dict2 = {3:30, 4:40}
dict3 = {5:50, 6:60}

dict4 = {}
for d in (dict1,dict2,dict3):
    dict4.update(d)
print(dict4)

OUTPUT:
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

--------------------------------------------------------------------------------
------------
```

Prac_5c
Write a Python program to sum all the items in a dictionary.
CODE:

```
#Method 1
dict1 = {'Python':90, 'C++':100, 'Java':80, 'C':50}
print(sum(dict1.values()))



#Method 2
dict2 = {'Python':90, 'C++':100, 'Java':80, 'C':50}
sum=0
for i in dict2.values():
    sum = sum+i
print(sum)
```

OUTPUT:

```
320

320
```

Prac_6a
Write a Python program to read an entire text file.
CODE:

```
f = open("myfile.txt","r")
print(f.read())
f.close()
```

OUTPUT:

```
one
two
three
four
five
six
seven
eight
nine
ten
```

Prac_6b
Write a Python program to append text to a file and display the text.
CODE:

```
f = open("myfile.txt","a")
f.write("\nEleven")
f.close()
```

OUTPUT:

```
one
two
three
four
five
six
seven
eight
nine
ten
Eleven
```

Prac_6c

Write a Python program to read last n lines of a file.
CODE:

```
n = int(input("Enter n lines: "))
f = open("myfile.txt","r")
for line in (f.readlines() [-n:]):
    print(line,end="")
f.close()
```

OUTPUT:

```
Enter n lines: 1
Eleven
```

Prac_7a
Design a class that store the information of student and display the same.
CODE:

```
class Student:
    def __init__(self, rollno, name, age, phone):
        self.rollno = rollno
        self.name = name
        self.age = age
        self.phone = phone

    def display(self):
        print("Student Roll No = ",self.rollno)
        print("Student Name = ",self.name)
        print("Student Age = ",self.age)
        print("Student Phone Number = ",self.phone)

print("----------- Enter Student Details -------------")
rollno = int(input("Enter your roll no: "))
name = input("Enter your name: ")
age = int(input("Enter your age: "))
phone = int(input("Enter your phone: "))

ob = Student(rollno, name, age, phone)
print("--------- Display Student Details ----------")
ob.display()
```

OUTPUT:

```
----------- Enter Student Details -------------
Enter your roll no: 45
Enter your name: Rahul Gupta
Enter your age: 19
Enter your phone: 8930224560
--------- Display Student Details ----------
Student Roll No =  45
Student Name =  Rahul Gupta
Student Age =  19
Student Phone Number =  8930224560
```