1. Install Python Interpreter and use it to perform different Mathematical Computations. Try to do all the operations present in a Scientific Calculator

Source code:

```
import math
import random

# Function to add two numbers
def add(num1, num2):
    print(num1, "+", num2, "=",num1+num2)

# Function to subtract two numbers
def subtract(num1, num2):
    print(num1, "-", num2, "=",num1-num2)

# Function to multiply two numbers
def multiply(num1, num2):
    print(num1, "*", num2, "=",num1*num2)

# Function to divide two numbers
def divide(num1, num2):
    print(num1, "/", num2, "=",num1/num2)

# Function to modulous division two numbers
def modulodiv(num1, num2):
    print(num1, "%", num2, "=",num1%num2)

# Function to calculate the powers of 10
def powers_of_10(num1):
    print("10", "^", num1, "=",pow(10,num1))

# Function to calculate the square of a number
def square(num1):
    print(num1, "^","2" , "=",pow(num1,2))

# Function to calculate the cube of a number
def cube(num1):
    print(num1, "^","3" , "=",pow(num1,3))

# Function to calculate the power of y with x
def power_of_xy(num1,num2):
    print(num1, "^",num2 , "=",pow(num1,num2))

# Function to calculate the squareroot of a number
def squareroot(num1):
    print("Square Root of ",num1,"=",math.sqrt(num1))
```

```
# Function to calculate the Cuberoot of a number
def cuberoot(num1):
    print("Cube Root of ",num1,"=",pow(num1,(1/3)))

# Function to calculate the sine (x) in degrees
def sinx(num1):
    print("Sine(",num1,"Degrees) =",math.sin(math.radians(num1)))

# Function to calculate the cosine of a number
def cosx(num1):
    print("Cosine(",num1,"Degrees) =",math.cos(math.radians(num1)))

# Function to calculate the tan of a number
def tanx(num1):
    print("Tan(",num1,"Degrees) =",math.tan(math.radians(num1)))

# Function to calculate the natural logarithm (Ln) of a number
def natural_log(num1):
    print("The natural logarithm (Ln) of ",num1,"=",math.log(num1))

# Function to calculate the Logarithm of base 10 of a number
def log(num1):
    print("The Logarithm of base 10 of ",num1,"=",math.log10(num1))

# Function to calculate the Factorial of a number
def factorial(num1):
    print("The Factorial of ",num1,"=",math.factorial(num1))

# Function to calculate the Roundoff of a number
def roundoff(num1):
    print("The Roundoff of ",num1,"=",round(num1,2))

# Function to calculate the Floor of a number
def floor(num1):
    print("The Floor of ",num1,"=",math.floor(num1))

# Function to calculate the Ceil of a number
def ceil(num1):
    print("The Ceil of ",num1,"=",math.ceil(num1))

# Function to calculate the Absolute of a number
def absolute(num1):
    print("The Absolute of ",num1,"=",abs(num1))

# Function to calculate the value of PI
def PI_value():
    print("The value of PI =",math.pi)

# Function to calculate the Inverse of a number
def inverse(num1):
```

```
    print("The Inverse of ",num1,"=",(1/num1))

# Function to generate a Random number
def random(num1,num2):
    import random
    randomlist = random.sample(range(num1, num2), 10)
    print("The Generated 10 Random numbers between",num1,"and ",num2,"are
=",randomlist)

# Function to calculate the e^x of a number
def e_x(num1):
    print("The e^x of ",num1,"=",math.exp(num1))

# Fuction Calls

add(5,6)
subtract(10,5)
multiply(3,12)
divide(2,20)
modulodiv(9,34)
modulodiv(5,12)
powers_of_10(5)
square(12)
cube(6)
sinx(3)
cosx(5)
tanx(7)
natural_log(2)
log(3)
factorial(5)
roundoff(8)
floor(4)
ceil(7)
absolute(9)
PI_value()
inverse(12)
random(5,15)
e_x(7)
```

Output:

```
5 + 6 = 11
10 - 5 = 5
3 * 12 = 36
2 / 20 = 0.1
9 % 34 = 9
5 % 12 = 5
10 ^ 5 = 100000
12 ^ 2 = 144
6 ^ 3 = 216
```

Sine(3 Degrees) = 0.052335956242943835
Cosine(5 Degrees) = 0.9961946980917455
Tan(7 Degrees) = 0.1227845609029046
The natural logarithm (Ln) of 2 = 0.6931471805599453
The Logarithm of base 10 of 3 = 0.47712125471966244
The Factorial of 5 = 120
The Roundoff of 8 = 8
The Floor of 4 = 4
The Ceil of 7 = 7
The Absolute of 9 = 9
The value of PI = 3.141592653589793
The Inverse of 12 = 0.08333333333333333
The Generated 10 Random numbers between 5 and 15 are = [13, 14, 10, 6, 8, 5, 12, 7, 9, 11]
The e^x of 7 = 1096.6331584284585

2. Write a function that draws a grid like the following:

```
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
```

Source code:

```
def grid(row, col):
    x = ('+----' * col + '+')
    y = ('\n' +'|    ' *  (col+1))
    return ((x + 4*y) +'\n')*row + x
print(grid(2,2))
```

Output:

```
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
```

3. Write a function that draws a Pyramid with # symbols Up to 15 hashes at the bottom

Source code:

```
def pyramid(h):
    for i in range(h):
        print(" " * (h-i - 1) + "#" * (2 * i + 1))
pyramid(8)
```

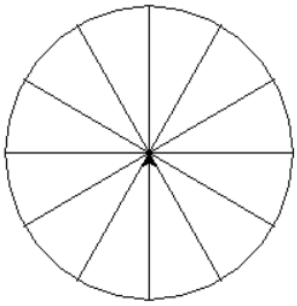
Output:

4. Using turtles concept draw a wheel of your choice

Source code:

```
import turtle as t
t.circle(100)
t.lt(90)
t.fd(100)
for i in range(12):
    t.fd(100)
    t.bk(100)
    t.lt(30)
```

Output:

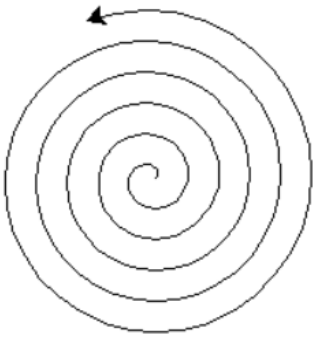


5. Write a program that draws Archimedean Spiral

Source Code:

```
import turtle as t
for i in range(100):
    t.circle(i,20)
```

Output:



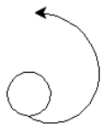
6. The letters of the alphabet can be constructed from a moderate number of basic elements, like vertical and horizontal lines and a few curves. Design an alphabet that can be drawn with a minimal number of basic elements and then write functions that draw the letters. The alphabet can belong to any Natural language excluding English. You should consider at least Ten letters of the alphabet.

Source Code:

```
import turtle
alex = turtle.Turtle()
```

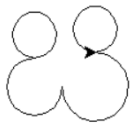
1.English ‘A’in telugu

```
alex.penup()
alex.goto(-200,200)
alex.down()
alex.circle(20)
alex.rt(35)
for i in range(35,62):
    alex.circle(i,8)
```



2.English B in telugu

```
alex.penup()
alex.fd(100)
alex.pendown()
alex.circle(20)
alex.rt(180)
alex.circle(25,180)
alex.circle(25,90)
alex.rt(180)
alex.circle(30,270)
alex.rt(180)
alex.circle(20)
```



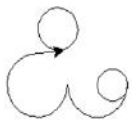
3.English C in telugu

```
alex.penup()
alex.fd(250)
alex.pendown()
alex.rt(270)
alex.circle(20,180)
alex.penup()
```

```
alex.lt(90)
alex.fd(40)
alex.pendown()
alex.lt(270)
alex.circle(20,180)
alex.circle(100,30)
alex.penup()
alex.fd(30)
alex.pendown()
alex.circle(20)
```

4.English D in telugu

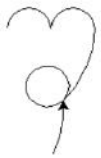
```
alex.penup()
alex.fd(330)
alex.pendown()
alex.lt(180)
alex.circle(30,270)
alex.rt(180)
alex.circle(30,180)
alex.circle(15)
alex.penup()
alex.goto(0,0)
alex.rt(90)
alex.fd(330)
alex.pendown()
alex.rt(360)
alex.circle(100,15)
alex.circle(20)
```



5.English E in telugu

```
alex.penup()
alex.fd(480)
alex.pendown()
alex.lt(90)
alex.circle(20,180)
```

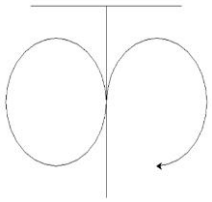
```
alex.lt(90)
alex.penup()
alex.fd(80)
alex.pendown()
alex.lt(90)
alex.circle(20,180)
alex.penup()
alex.lt(90)
alex.fd(40)
alex.pendown()
alex.lt(90)
alex.penup()
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,45)
alex.pendown()
alex.circle(100,45)
alex.penup()
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,45)
alex.pendown()
alex.circle(20)
alex.penup()
alex.lt(50)
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,90)
alex.circle(100,60)
alex.pendown()
alex.circle(100,30)
```



6.Hindi Ka

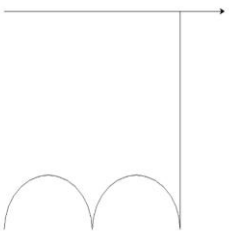
```
alex.rt(0)
alex.penup()
alex.goto(-150,300)
alex.pendown()
alex.fd(300)
```

```
alex.penup()
alex.goto(0,300)
alex.pendown()
alex.rt(90)
alex.fd(300)
alex.penup()
alex.goto(0,150)
alex.pendown()
alex.rt(180)
alex.circle(100)
alex.bk(0)
alex.rt(0)
alex.circle(-100,270)
```



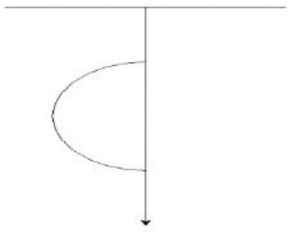
7.hindi "la"

```
alex.penup()
alex.goto(-500,-200)
alex.pendown()
alex.lt(270)
alex.circle(100,-180)
alex.lt(180)
alex.circle(100,-180)
alex.fd(400)
alex.penup()
alex.goto(-500,200)
alex.rt(90)
alex.pendown()
alex.fd(500)
```



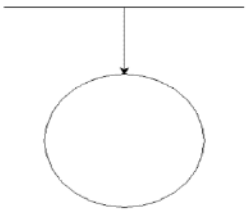
8.hindi "va"

```
alex.circle(100,-180)
alex.penup()
alex.goto(-150,300)
alex.rt(180)
alex.pendown()
alex.fd(300)
alex.penup()
alex.goto(0,300)
alex.rt(90)
alex.pendown()
alex.fd(400)
```



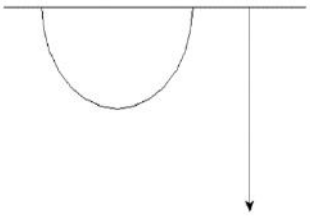
9.Hindi "Taa"

```
alex.circle(100)
alex.penup()
alex.goto(0,300)
alex.rt(0)
alex.goto(-150,300)
alex.pendown()
alex.fd(300)
alex.penup()
alex.goto(0,300)
alex.pendown()
alex.rt(90)
alex.fd(100)
```



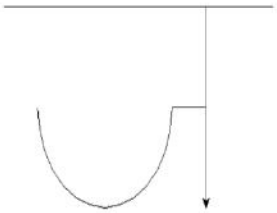
10.hindi "NAA"

```
alex.rt(-90)
alex.circle(100,-180)
alex.lt(90)
alex.penup()
alex.goto(-250,0)
alex.pendown()
alex.fd(400)
alex.penup()
alex.goto(75,0)
alex.pendown()
alex.lt(-90)
alex.fd(200)
```



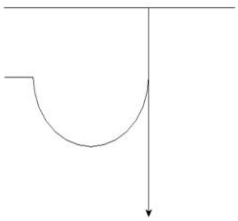
11.hindi "Ja"

```
alex.rt(-90)
alex.circle(100,-180)
alex.lt(90)
alex.penup()
alex.goto(0,0)
alex.pendown()
alex.fd(50)
alex.rt(0)
alex.penup()
alex.goto(-250,100)
alex.pendown()
alex.fd(400)
alex.penup()
alex.goto(50,100)
alex.rt(90)
alex.pendown()
alex.fd(200)
```



12.hindi "cha"

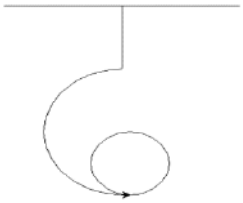
```
alex.rt(-90)
alex.circle(100,-180)
alex.lt(90)
alex.goto(-250,0)
alex.penup()
alex.goto(0,0)
alex.pendown()
#alex.fd(50)
alex.rt(0)
alex.penup()
alex.goto(-250,100)
alex.pendown()
alex.fd(400)
alex.penup()
alex.goto(0,100)
alex.rt(90)
alex.pendown()
alex.fd(300)
```



13.Hindi "six dha"

```
alex.circle(100,-180)
alex.penup()
alex.goto(-150,300)
alex.rt(180)
alex.pendown()
alex.fd(300)
alex.penup()
alex.goto(0,300)
alex.pendown()
```

```
alex.rt(90)
alex.fd(100)
alex.penup()
alex.goto(0,0)
alex.pendown()
alex.rt(-90)
alex.goto(10,0)
alex.circle(50)
```



14.Telugu “ruu”

```
#alex.bgcolor("black")
alex.pensize(4)
#alex.pencolor("white")
alex.pu()
alex.setpos(-100,0)
alex.pd()
alex.lt(90)
alex.circle(-20,180)
alex.circle(-40,60)
alex.circle(25,240)
alex.seth(270)
alex.circle(25,180)
alex.circle(100,45)
alex.pu()
alex.rt(180)
alex.circle(-100,45)
alex.pd()
alex.seth(270)
alex.circle(25,180)
alex.circle(100,45)
alex.pu()
alex.rt(180)
alex.circle(-100,45)
alex.pd()
alex.seth(270)
alex.circle(25,180)
alex.circle(100,45)
alex.seth(0)
alex.fd(70)
alex.circle(-15)
```

7. The time module provides a function, also named time that returns the current Greenwich Mean Time in "the epoch", which is an arbitrary time used as a reference point. On UNIX systems, the epoch is 1 January 1970.

>>> import time

>>>time.time()

1437746094.5735958

Write a script that reads the current time and converts it to a time of day in hours, minutes, and seconds, plus the number of days since the epoch.

Source Code:

```
import time
import datetime
from datetime import date
epoch=time.time()
today = date.today()
since=(datetime.datetime.utcnow() - datetime.datetime(1970,1,1)).days
h=time.strftime('%H', time.localtime(epoch))
mi=time.strftime('%M', time.localtime(epoch))
s=time.strftime('%S', time.localtime(epoch))
d=time.strftime('%d', time.localtime(epoch))
mo=time.strftime('%m', time.localtime(epoch))
y=time.strftime('%Y', time.localtime(epoch))
f=time.strftime('%d-%m-%Y %H:%M:%S', time.localtime(epoch))
d1 = today.strftime("%d/%m/%y")
d2 = today.strftime("%d/%m/%Y")
d3 = today.strftime("%d-%b-%Y")
d4 = today.strftime("%d %B %Y")
print("Epoch time:",epoch)
print("Number of days since January 1, 1970:",since,"Days")
print("Day:",d)
print("Month:",mo)
print("Year:",y)
print("Hours:",h)
print("Minutes:",mi)
print("Seconds:",s)
```

```
print("DD/MM/YY:", d1)
print("DD/MM/YYYY:", d2)
print("DD-MMM-YYYY:", d3)
print("Day Month Year:", d4)
print("Todays date and time:",f)
```

Output:

```
Epoch time: 1639244594.8350816
Number of days since January 1, 1970: 18972 Days
Day: 11
Month: 12
Year: 2021
Hours: 23
Minutes: 13
Seconds: 14
DD/MM/YY: 11/12/21
DD/MM/YYYY: 11/12/2021
DD-MMM-YYYY: 11-Dec-2021
Day Month Year: 11 December 2021
Todays date and time: 11-12-2021 23:13:14
```

8. Given $n+r+1 = 2^r$,n is the input and r is to be determined. Write a program which computes minimum value of r that satisfies the above.

Source Code:

```
import math
start=-1000
stop=1000
step=1
for n in range(start,stop,step):
    for r in range(start,stop,step):
        x=n+r+1
        y=pow(2,r)
        if(x==y):
            print("Value of n:",n,"Value of r:",r,"Value of ",pow(2,r))
```

Output:

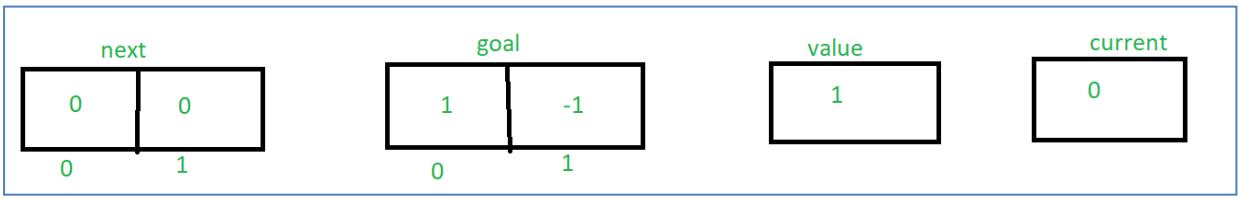
```
Value of n: 0 Value of r: 0 Value of  1
Value of n: 0 Value of r: 1 Value of  2
Value of n: 1 Value of r: 2 Value of  4
Value of n: 4 Value of r: 3 Value of  8
Value of n: 11 Value of r: 4 Value of  16
Value of n: 26 Value of r: 5 Value of  32
Value of n: 57 Value of r: 6 Value of  64
Value of n: 120 Value of r: 7 Value of  128
Value of n: 247 Value of r: 8 Value of  256
Value of n: 502 Value of r: 9 Value of  512
```

9. Write a program that evaluates Ackermann function

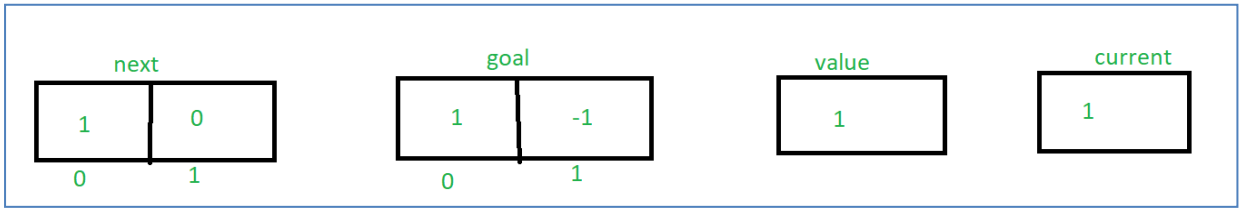
Ackermann function is defined as:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

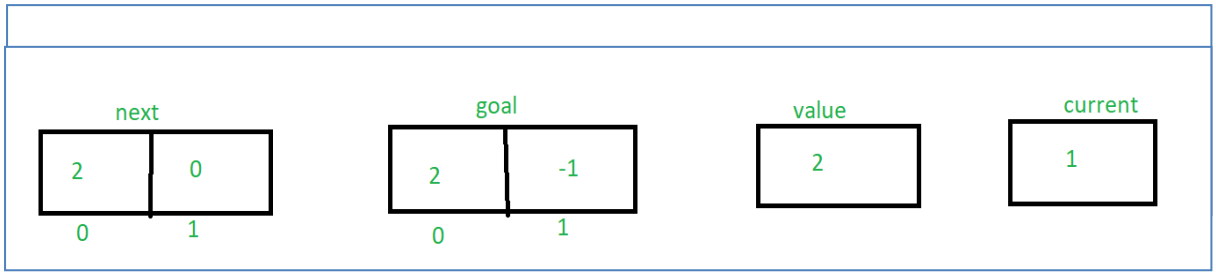
Here's the explanation of the given Algorithm:
Let me explain the algorithm by taking the example A(1, 2) where m = 1 and n = 2
So according to the algorithm initially the value of **next, goal, value and current** are:



Though next[current] != goal[current], so else statement will execute and transferring become false.
So now, the value of next, goal, value and current are:



Similarly by tracing the algorithm until next[m] = 3 the value of next, goal, value and current are changing accordingly. Here's the explanation how the values are changing,



Finally returning the value e.g 4

Solve A(1, 2)?

Answer:

Given problem is A(1, 2)
Here m = 1, n = 2 e.g m > 0 and n > 0
Hence applying third condition of Ackermann function
A(1, 2) = A(0, A(1, 1)) ———— (1)
Now, Let's find A(1, 1) by applying third condition of Ackermann function
A(1, 1) = A(0, A(1, 0)) ———— (2)
Now, Let's find A(1, 0) by applying second condition of Ackermann function
A(1, 0) = A(0, 1) ———— (3)
Now, Let's find A(0, 1) by applying first condition of Ackermann function
A(0, 1) = 1 + 1 = 2
Now put this value in equation 3
Hence A(1, 0) = 2
Now put this value in equation 2
A(1, 1) = A(0, 2) ———— (4)
Now, Let's find A(0, 2) by applying first condition of Ackermann function
A(0, 2) = 2 + 1 = 3
Now put this value in equation 4
Hence A(1, 1) = 3
Now put this value in equation 1
A(1, 2) = A(0, 3) ———— (5)
Now, Let's find A(0, 3) by applying first condition of Ackermann function
A(0, 3) = 3 + 1 = 4
Now put this value in equation 5
Hence A(1, 2) = 4
So, A (1, 2) = 4

Question: Solve A(2, 1)?

Answer: 5

Question: Solve A(2, 2)?

Answer: 7

Source Code:

```
def ackermann(m,n):
    if m == 0:
        return (n + 1)
    elif n == 0:
        return ackermann(m - 1, 1)
    else:
        return ackermann(m - 1, ackermann(m, n - 1))

x=int(input("value for m "))
y=int(input("value for n "))
print("The value for m:",x)
print("The value for n:",y)
print("Result of your inputs according to the Ackermann Function is:")
print(ackermann(x, y))
```

Output:

```
value for m 2
value for n 2
The value for m: 2
The value for n: 2
Result of your inputs according to the Ackermann Function is:
7
```

10. The mathematician SrinivasaRamanujan found an infinite series that can be used to generate a numerical approximation of 1/ π :

Write a function called estimate_pi that uses this formula to compute and return an estimate of π.

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

It should use a while loop to compute terms of the summation until the last term is smaller than $1e^{-15}$ (which is Python notation for 10^{-15}). You can check the result by comparing it to math.pi.

Source Code:

```
import math
def estimate_pi():
```

```
    k=0.0
    last_term=1.0
    sigma=0
    while last_term > 1e-15:

last_term=((math.factorial(4.0*k))*(1103.0+26390.0*(k)))/((math.factorial(k)**4.0)*(396.0*
*(4.0*k)))
        k+=1.0
        sigma+=last_term
        result=((2*math.sqrt(2))/9801)*sigma
        return 1/result

print("Ramanujam PI value:",estimate_pi())
print("math.PI value:",math.pi)
print("22/7 value:",22/7)
```

Output:

```
Ramanujam PI value: 3.141592653589793
math.PI value: 3.141592653589793
22/7 value: 3.142857142857143
```

11. Choose any five built-in string functions of C language. Implement them on your own in Python. You should not use string related Python built-in functions.

Understanding slice notation
a[start:stop] # items start through stop-1
a[start:]   # items start through the rest of the array
a[:stop]   # items from the beginning through stop-1
a[:]   # a copy of the whole array
There is also the step value, which can be used with any of the above:
a[start:stop:step] # start through not past stop, by step
a[-1]   # last item in the array
a[-2:]   # last two items in the array
a[:-2]   # everything except the last two items
a[::-1]   # all items in the array, reversed
a[1::-1]   # the first two items, reversed
a[:-3:-1] # the last two items, reversed
a[-3::-1] # everything except the last two items, reversed

Source Code:

1. Upper Case

```
str_data=input("Enter a string in Lower case:")
def uppercase(str_data):
    result = ''
```

```
    for char in str_data:
        if ord(char) >= 65:
            result += chr(ord(char) - 32)
    return result
print(uppercase(str_data))
```

2. Lower Case

```
str_data=input("Enter a string in Upper case:")
def lowercase(str_data):
    result = ''
    for char in str_data:
        if ord(char) >= 65:
            result += chr(ord(char) + 32)
    return result
print(lowercase(str_data))
```

3. String Length

```
string=input("Enter a string:")
count=0
for i in string:
    count=count+1
print("Length of the string is:", count)
```

4. Counting the number of occurance of a character in a string

```
word=input("Enter a string:")
l=input("Enter a character to search:")
count = 0
for letter in word:
    if (letter==l):
        count = count + 1
print("The Total Number of Occurances of letter:",l,"\nin the String:",word,"\nis:",count)
```

5. String Concatenation

```
word1=input("Enter the first string:")
word2=input("Enter the Second string:")
word3=word1+word2
print("The First String:",word1,"\nThe Second String:",word2,"\nThe Concatenated String is:",word3)
```

6. Checking and Arranging the strings in Alphabetical Order

```
words = ["Bombay", "Delhi", "Calcutta","Chennai"]
check = "Chennai"
```

```
for c in words:
   if c < check:
         print ("%s comes before %s" % (c, check))
   elif c > check:
         print ("%s comes after %s" % (c, check))
   else:
         print ("%s is similar to %s" % (c, check))
```

7. Palindrome

```
mystr = input("Enter the string: ")
if str(mystr) == str(mystr)[::-1]: #checking using slicing opearator
  print("%s is a Palindrome"%mystr)
else:
  print("%s is NOT a Palindrome"%mystr)
```

(or)

```
string=input("Enter the String:")
count=0
for i in string:
     count=count+1
c=int(count/2)
flag=0

for i in range(c):
   if string[i]!=string[count-i-1]:
      flag=1
      break

if flag==1:
   print ("%s is NOT a Palindrome"%string)
else:
   print ("%s is a Palindrome"%string)
```

Output:

```
Enter a string in Lower case:python
PYTHON
Enter a string in Upper case:PYTHON
python
Enter a string:python programming
Length of the string is: 18
Enter a string:python programs
Enter a character to search:o
The Total Number of Occurances of letter: o
in the String: python programs
is: 2
Enter the first string:Hello
Enter the Second string:python
The First String: Hello
The Second String: python
The Concatenated String is: Hellopython
Bombay comes before Chennai
Delhi comes after Chennai
Calcutta comes before Chennai
Chennai is similar to Chennai
Enter the string: python
python is NOT a Palindrome
Enter the String:343
343 is a Palindrome
```

12.Given a text of characters, Write a program which counts number of vowels, consonants and special characters.

Source Code:

```
str = input('Enter the string : ')
vowels = 0
digits = 0
consonants = 0
spaces = 0
symbols = 0
str = str.lower()
for i in range(0, len(str)):
    if(str[i] == 'a' or str[i] == 'e' or str[i] == 'i'or str[i] == 'o' or str[i] == 'u'):
        vowels = vowels + 1
    elif((str[i] >= 'a'and str[i] <= 'z')):
        consonants = consonants + 1
    elif( str[i] >= '0' and str[i] <= '9'):
        digits = digits + 1
    elif (str[i] ==' ' or str[i]==' '):
        spaces = spaces + 1
    else:
        symbols = symbols + 1
print('Vowels:', vowels);
print('Consonants:', consonants);
print('Digits:', digits);
print('White spaces:', spaces);
```

```
print('Symbols :', symbols);
```

Output:

```
Enter the string : ** Annamacharya Institute of Technology 123 **
Vowels: 13
Consonants: 20
Digits: 3
White spaces: 6
Symbols : 4
```

13. Given a word which is a string of characters. Given an integer say ‘n’, rotate each character by ‘n’ positions and print it. Note that ‘n’ can be positive or negative.

Source Code:

```
print("If the number is +ve the rotation will be left and right")
print("If the number is -ve the rotation will be right and left")
string=input('Enter the string:')
rotate=int(input('Enter the number of characters to rotate:'))

# In-place rotates s towards left by d
def leftrotate(s, d):
    tmp = s[d : ] + s[0 : d]
    return tmp

# In-place rotates s
# towards right by d
def rightrotate(s, d):

   return leftrotate(s, len(s) - d)

print("Left rotation string is:",leftrotate(string, rotate))
print("Right rotation string is:",rightrotate(string, rotate))
```

(or)

```
print("If the number is +ve the rotation will be left and right")
print("If the number is -ve the rotation will be right and left")
string=input('Enter the string:')
rot=int(input('Enter the number of characters to rotate:'))
def rotate(input,d):
   # Slice string in two parts for left and right
   Lfirst = input[0 : d]
   Lsecond = input[d :]
```

```
    Rfirst = input[0 : len(input)-d]
    Rsecond = input[len(input)-d : ]
    print ("Left Rotation : ", (Lsecond + Lfirst) )
    print ("Right Rotation : ", (Rsecond + Rfirst) )
rotate(string,rot)
```

Output:

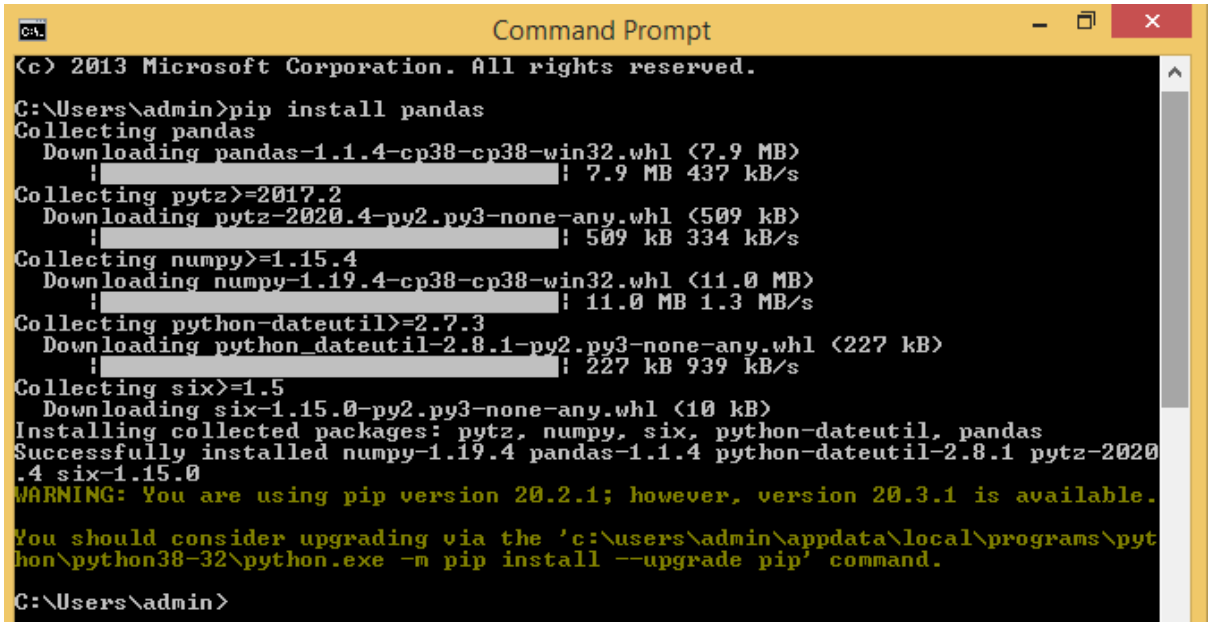
```
If the number is +ve the rotation will be left and right
If the number is -ve the rotation will be right and left
Enter the string:mufasa is the father of simba in lion king
Enter the number of characters to rotate:5
Left rotation string is: a is the father of simba in lion kingmufas
Right rotation string is:  kingmufasa is the father of simba in lion
```

```
If the number is +ve the rotation will be left and right
If the number is -ve the rotation will be right and left
Enter the string:mufasa is the father of simba in lion king
Enter the number of characters to rotate:-3
Left rotation string is: ingmufasa is the father of simba in lion k
Right rotation string is: mufasa is the father of simba in lion king
```

14. Given rows of text, write it in the form of columns.

First install pandas

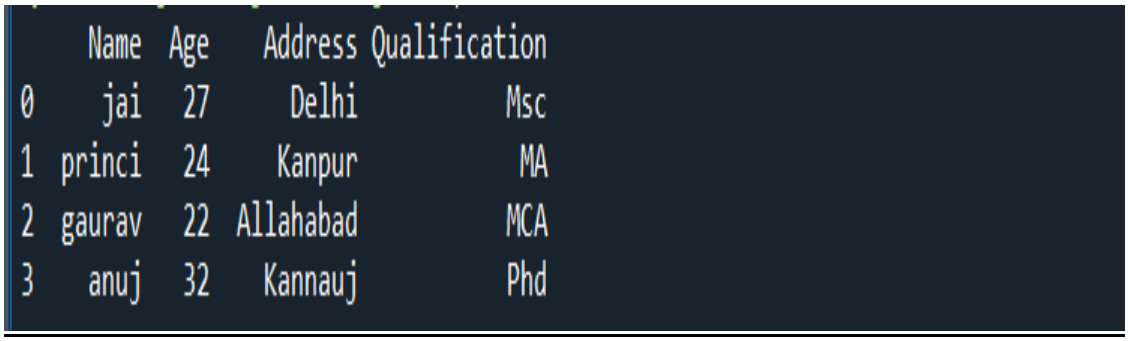
1. Go to RUN and Type cmd
2. Type pip install pandas
3. Once installed execute the program



Source Code:

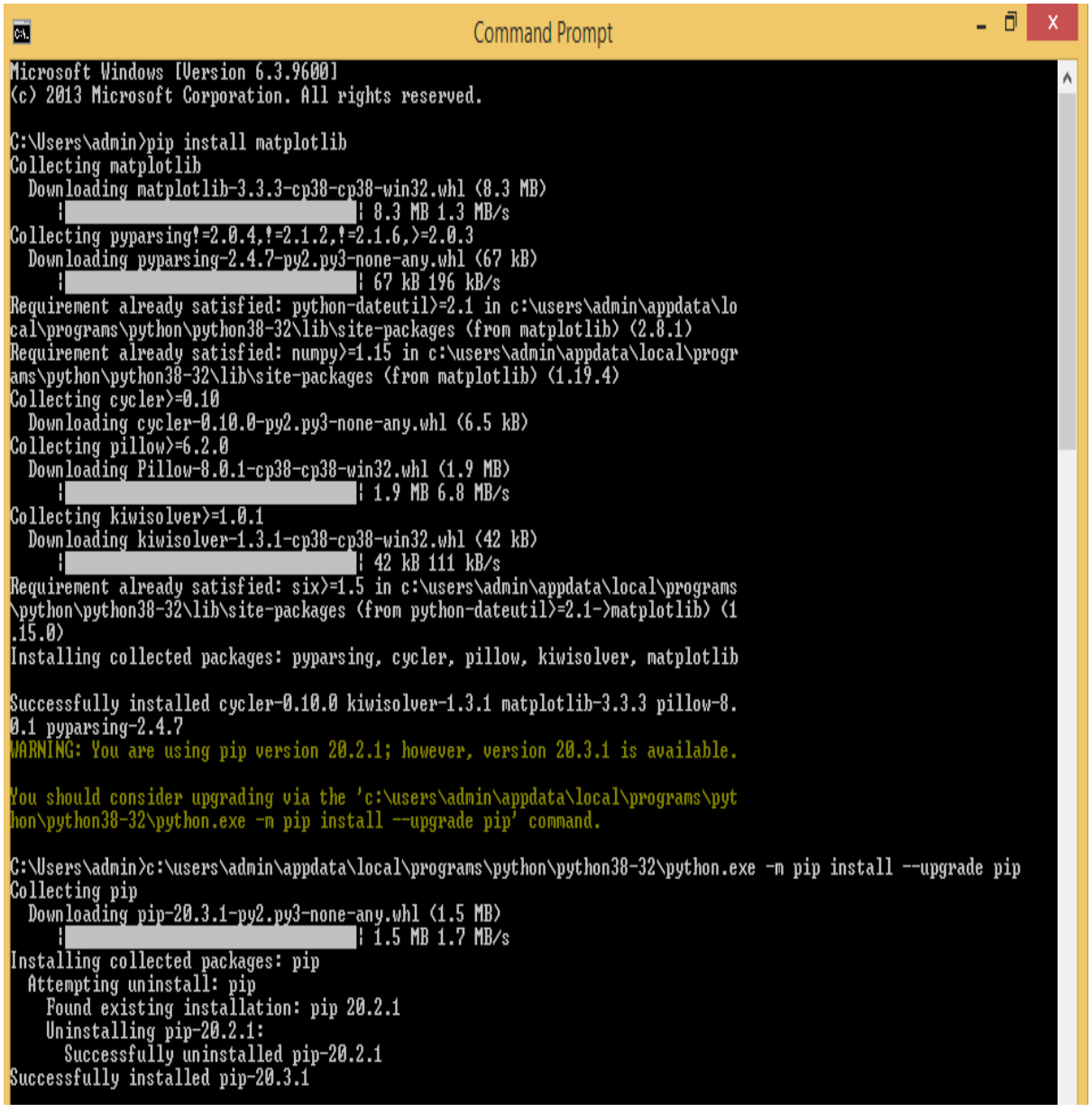
```
# Import pandas package
import pandas as pd

# Define a dictionary containing employee data
data = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# converting and overwriting values in column
df["Name"]= df["Name"].str.lower()

print(df)
```

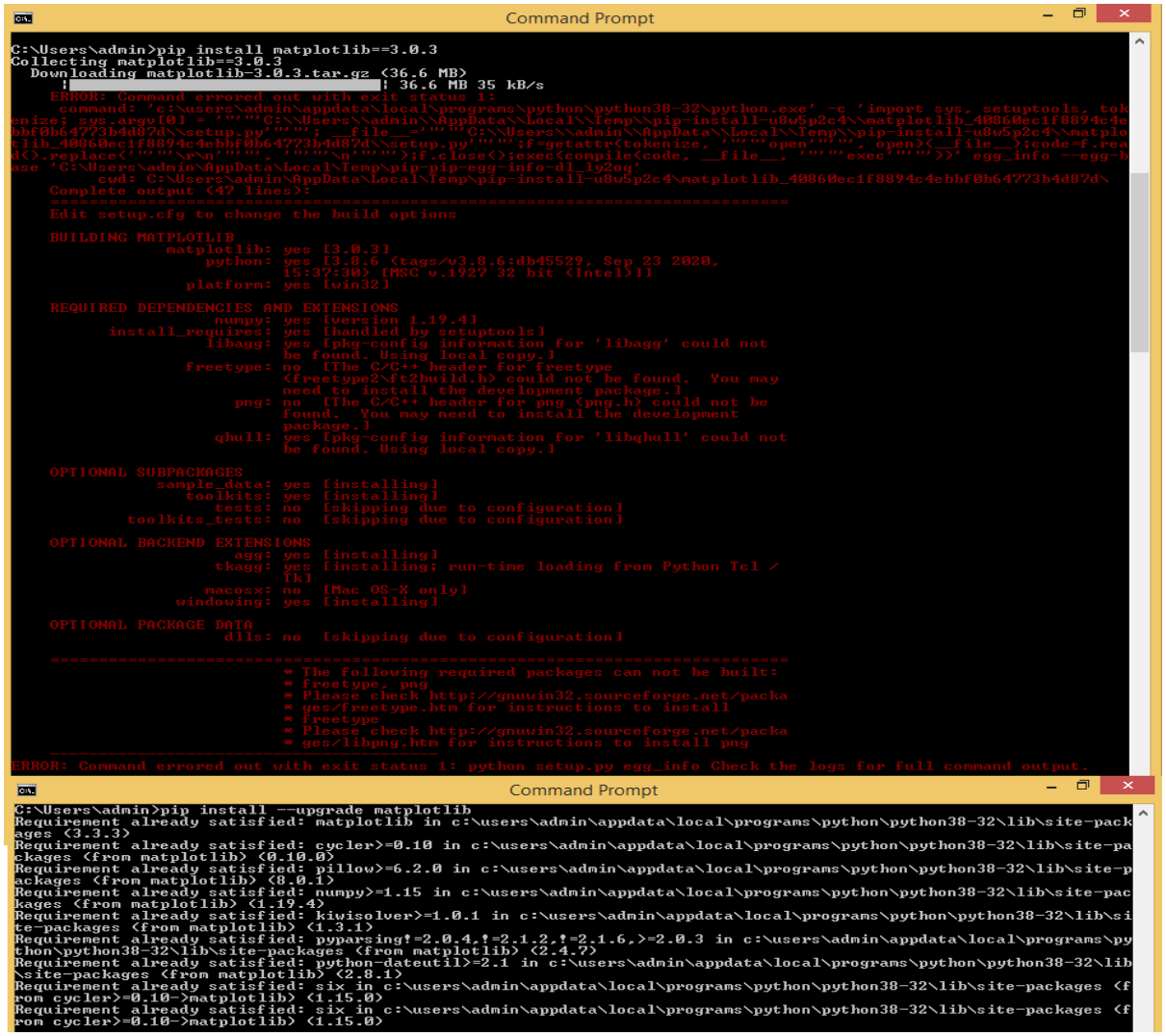
Output:



	Name	Age	Address	Qualification
0	jai	27	Delhi	Msc
1	princi	24	Kanpur	MA
2	gaurav	22	Allahabad	MCA
3	anuj	32	Kannauj	Phd

15.Given a page of text. Count the number of occurrences of each latter (Assume case insensitivity and don't consider special characters). Draw a histogram to represent the same print("The Total Number of characters with frequecies:\n",freqs)





Steps for Installing matplotlib module

1. Go to command prompt in windows
2. Type **pip install matplotlib**
3. Type **pip install matplotlib==3.0.3**
4. Type **pip install --upgrade matplotlib**

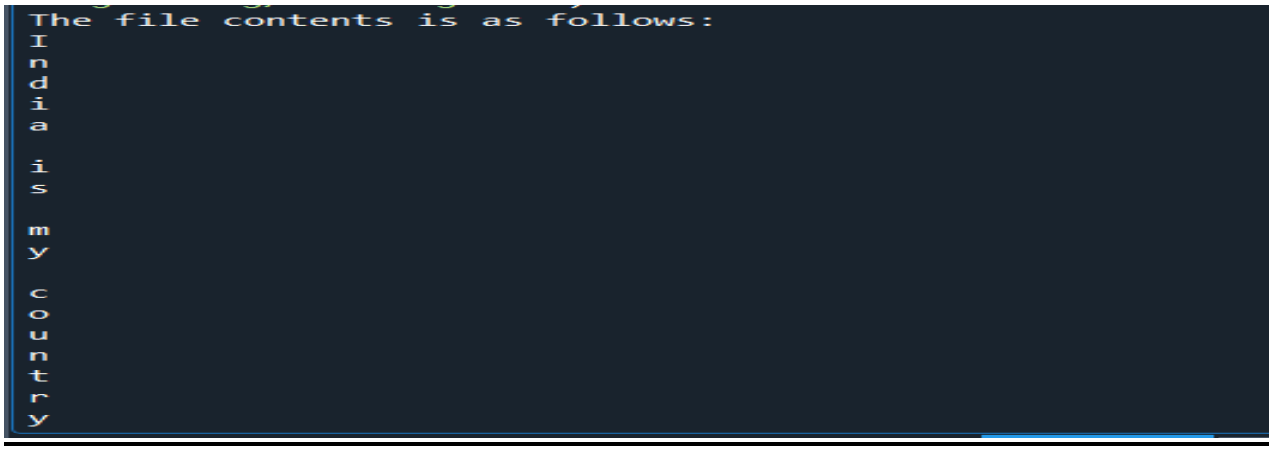
words.txt (Input File)

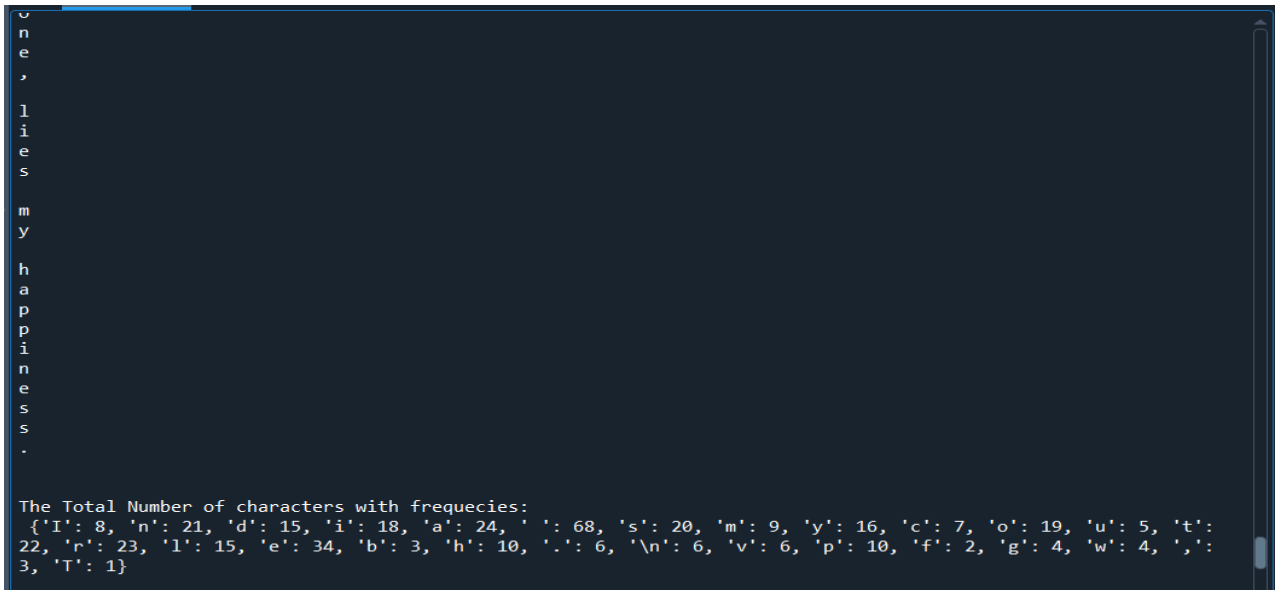
India is my country and all Indians are my brothers and sisters.
I love my country and I am proud of its rich and varied heritage.
I shall always strive to be worthy of it. I shall give respect to my parents,
teachers and elders and treat everyone with courtesy.
To my country and my people, I pledge my devotion.
In their well being and prosperity alone, lies my happiness.

Source Code:

```
freqs = {}
print("The file contents is as follows:")
with open('words.txt') as f:
    for line in f:
        for char in line:
            print(char)
            if char in freqs:
                freqs[char] += 1
            else:
                freqs[char] = 1
print("The Total Number of characters with frequecies:\n",freqs)
```

Output:





16. Write program which performs the following operations on list's. Don't use built-in functions
a) Updating elements of a list
b) Concatenation of list's
c) Check for member in the list
d) Insert into the list
e) Sum the elements of the list
f) Push and pop element of list
g) Sorting of list
h) Finding biggest and smallest elements in the list
i) Finding common elements in the list

Source Code:

a) Updating elements of a list

```
# Insert

a = []
b = []
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
a1=len(a)
pos=int(input("Enter the position to insert the element into the list:"))
c=[int(input("Enter the element to insert:"))]
pos=pos-1
if pos>=a1:
   print("The position is out of range")
else:
   print("The Inserted element is:",a[pos])
   print("The updated list is:")
   b=a[0:pos]+c+a[pos:]
   print(b)
```

Output:

```
Enter the list elements using spacebar: 1 2 3 4 5 6 7 8 9
Enter the position to insert the element into the list:2
Enter the element to insert:5
The Inserted element is: 2
The updated list is:
[1, 5, 2, 3, 4, 5, 6, 7, 8, 9]
```

Delete

```
a = []
b = []
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
a1=len(a)
pos=int(input("Enter the position to delete from the list :"))
pos=pos-1
if pos>=a1:
   print("The position is out of range")
else:
   print("The deleted element is:",a[pos])
   print("The updated list after deletion is:")
   b=a[0:pos]+a[pos+1:]
   print(b)
```

Output:

```
Enter the list elements using spacebar: 1 2 3 4 5 6 7 8 9
Enter the position to delete from the list :4
The deleted element is: 4
The updated list after deletion is:
[1, 2, 3, 5, 6, 7, 8, 9]
```

#b) Concatenation of list's

```
a = []
b = []
a = [int(item) for item in input("Enter the elements using spacebar for first list items : ").split()]
b = [int(item) for item in input("Enter the elements using spacebar for Second list items : ").split()]
c=a+b
print("List 1 is:",a)
print("List 2 is:",b)
print("The concatinated list is :",c)
```

Output:

```
Enter the elements using spacebar for first list items : 1 2 3 4
Enter the elements using spacebar for Second list items : 5 6 7 8
List 1 is: [1, 2, 3, 4]
List 2 is: [5, 6, 7, 8]
The concatinated list is : [1, 2, 3, 4, 5, 6, 7, 8]
```

#c) Check for member in the list

```
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
key=int(input("Enter the element for searching:"))
print("The Given List is:",a)
def common_data(lis,k):
    f=0
    # traverse in the 1st list
    for x in range(0,len(lis)):
        if lis[x] == k:
            f=1
            print("The searched element is present in the location:",x+1)
    if f==0:
        print("The entered element is not in the list")
common_data(a,key)
```

Output:

```
Enter the list elements using spacebar: 1 2 3 4 5 6 7 8 9
Enter the element for searching:6
The Given List is: [1, 2, 3, 4, 5, 6, 7, 8, 9]
The searched element is present in the location: 6
```

#d) Insert into the list

```
a = []
b = []
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
a1=len(a)
pos=int(input("Enter the position to insert the element into the list:"))
c=[int(input("Enter the element to insert:"))]
pos=pos-1
if pos>=a1:
    print("The position is out of range")
else:
    print("The Inserted element is:",a[pos])
    print("The updated list is:")

    b=a[0:pos]+c+a[pos:]
    print(b)
```

Output:

```
============================================================ R
Enter the list elements using spacebar: 1 2 3 4 5 6 7 8
Enter the position to insert the element into the list:5
Enter the element to insert:9
The Inserted element is: 5
The updated list is:
[1, 2, 3, 4, 9, 5, 6, 7, 8]
```

#e) Sum the elements of the list

```
a = []
a = [int(item) for item in input("Enter the list elements using spacebar : ").split()]
def add():
    total=0
    for x in a:
        total+=x
    return total
print("The Given List is:",a)
print("The Sum of elements in the list is:",add())
```

Output:

```
Enter the list elements using spacebar : 1 2 3 4 5 6
The Given List is: [1, 2, 3, 4, 5, 6]
The Sum of elements in the list is: 21
```

#f) Push and pop element of list

#pop

```
a = []
b = []
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
a1=len(a)
pos=int(input("Enter the position to pop from the list:"))
pos=pos-1
if pos>=a1:
    print("The position is out of range")
else:
    print("The popped element is:",a[pos])
    print("the updated list after pop is:")
    b=a[0:pos]+a[pos+1:]
    print(b)
```

Output:

```
Enter the list elements using spacebar: 2 3 4 5 6 7 8
Enter the position to pop from the list:3
The popped element is: 4
the updated list after pop is:
[2, 3, 5, 6, 7, 8]
```

```
# Push

a = []
b = []
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]
a1=len(a)
pos=int(input("Enter the position to push in the list:"))
c=[int(input("Enter the element to push:"))]
pos=pos-1
if pos>=a1:
    print("The position is out of range")
else:
    print("The pushed element is:",a[pos])
    print("The updated list is:")
    b=a[0:pos]+c+a[pos:]
    print(b)
```

Output:

```
Enter the list elements using spacebar : 6 5 8 4 2 3 9 1 5
Element After Sorting List in Ascending Order is :  [1, 2, 3, 4, 5, 5, 6, 8, 9]
```

```
#g) Sorting of list

a = [int(item) for item in input("Enter the list elements using spacebar : ").split()]

for i in range (len(a)):
    for j in range(i + 1, len(a)):
        if(a[i] > a[j]):
            temp = a[i]
            a[i] = a[j]
```

```
            a[j] = temp

print("Element After Sorting List in Ascending Order is : ", a)
```

Output:

```
Enter the list elements using spacebar: 1 3 2 5 6 7 4 8
The Biggest element in the list: 8
The Smallest element in the list: 1
```

#h) Finding biggest and smallest elements in the list

```
a = [int(item) for item in input("Enter the list elements using spacebar: ").split()]

for i in range (len(a)):
    for j in range(i + 1, len(a)):
        if(a[i] > a[j]):
            temp = a[i]
            a[i] = a[j]
            a[j] = temp

print("The Biggest element in the list:",a[len(a)-1])
print("The Smallest element in the list:",a[0])
```

Output:

```
Enter the list elements using spacebar: 1 3 4 5 2 7 6 8 9
The Biggest element in the list: 9
The Smallest element in the list: 1
```

#i) Finding common elements in the list

```
def common_data(list1, list2):
    f=0
    count=0
    # traverse in the 1st list
    for x in list1:
        # traverse in the 2nd list
        for y in list2:
            # if one common
            if x == y:
                f=1
```

```
            count+=1
            if(count==1):
                print("The common element in both the list are:")
            print(x)
    if(f==0):
        print("There is no common element in both the lists")
a = [int(item) for item in input("Enter the elements for first list items using spacebar: ").split()]
b = [int(item) for item in input("Enter the elements for Second list items using spacebar : ").split()]
print("The Given List1:",a)
print("The Given List2:",b)
common_data(a,b)
```

Output:

```
Enter the elements for first list items using spacebar: 2 4 3 5 6 1 8 9 5
Enter the elements for Second list items using spacebar : 5 4 3 2 7 6 9 8
The Given List1: [2, 4, 3, 5, 6, 1, 8, 9, 5]
The Given List2: [5, 4, 3, 2, 7, 6, 9, 8]
The common element in both the list are:
2
4
3
5
6
8
9
5
```

17.Write a program to count the number of vowels in a word.

Source Code:

```
string1 = input('Enter the string : ')
vowels = 0
string = string1.lower()
for i in range(0, len(string)):
    if(string[i] == 'a' or string[i] == 'e' or string[i] == 'i'or string[i] == 'o' or string[i] == 'u'):
        vowels = vowels + 1
print('The Given String:\n',string1)
print('Total number of Vowels are:',vowels)
```

Output:

```
Enter the string : What we think, we become.
The Given String:
 What we think, we become.
Total number of Vowels are: 7
```

18. Write a program that reads a file, breaks each line into words, strips whitespace and punctuation from the words, and converts them to lowercase.

Source Code:

```
import string
print ("The list of punctutation marks are:",string.punctuation)
from string import punctuation, whitespace

book = 'words.txt'

with open(book, 'r') as fd:
    words = fd.read().split()

#remove punctuation, whitespace, uppercase
def clean(word):
    cleansed = ''
    for char in word:
        if ((char in punctuation) or (char in whitespace)):
            pass
        else:
            cleansed += char.lower()
    print(cleansed)
    return cleansed

print ("{} has {} 'words'".format(book, len([clean(word) for word in words])))
```

Output:

```
The list of punctutation marks are: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
india
is
my
country
and
all
indians
are
my
brothers
and
sisters
i
love
my
country
and
i
am
proud
of
its
rich
and
varied
heritage
i
shall
always
strive
to
be
worthy
of
it
i
```

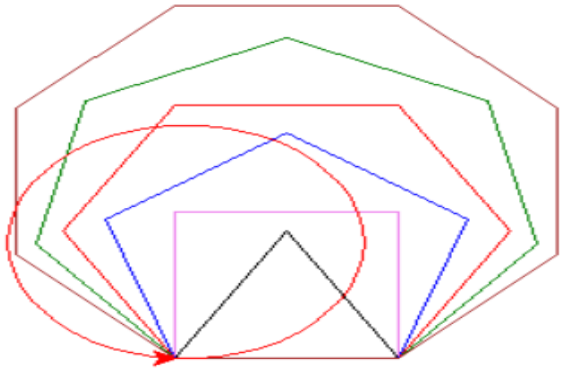
```
of
it
i
shall
give
respect
to
my
parents
teachers
and
elders
and
treat
everyone
with
courtesy
to
my
country
and
my
people
i
pledge
my
devotion
in
their
well
being
and
prosperity
alone
lies
my
happiness
words.txt has 70 'words'
```

19. Consider turtle object. Write functions to draw triangle, rectangle, polygon, circle and sphere. Use object oriented approach.

Source Code:

```
import turtle
bob=turtle.Turtle()
def polygon(t, sides, length):
  angle = 360.0 / sides #Float value at Numerator
  for i in range(sides):
    t.fd(length)
    t.lt(angle)
turtle.delay(40)
bob.color("black")
polygon(bob, sides=3, length=100) #Triangle as keyword arguments
bob.color("violet")
polygon(bob, 4, 100)#Square
bob.color("blue")
polygon(bob, 5, 100)#Pentagon
bob.color("red")
polygon(bob, 6, 100)#Hexagon
bob.color("green")
polygon(bob, 7, 100)#Septagon
bob.color("brown")
polygon(bob, 8, 100)#Octagon
bob.color("red")
polygon(bob, 50, 10)#Octagon
```

Output:



20. Write a program illustrating the object oriented features supported by Python.

Source Code:

```
class parrot:
    species="bird"
    def __init__(s,name,age):
        s.name=name
        s.age=age
blu=parrot("blu",10)
woo=parrot("woo",15)
print("blu is a {}".format(blu.__class__.species))
print("woo is a {}".format(woo.__class__.species))
print("{} is {} years old".format(blu.name,blu.age))
print("{} is {} years old".format(woo.name,woo.age))
```

Output:

```
blu is a bird
woo is a bird
blu is 10 years old
woo is 15 years old
```

21. Design a Python script using the Turtle graphics library to construct a turtle bar chart representing the grades obtained by N students read from a file categorizing them into distinction, first class, second class, third class and failed.

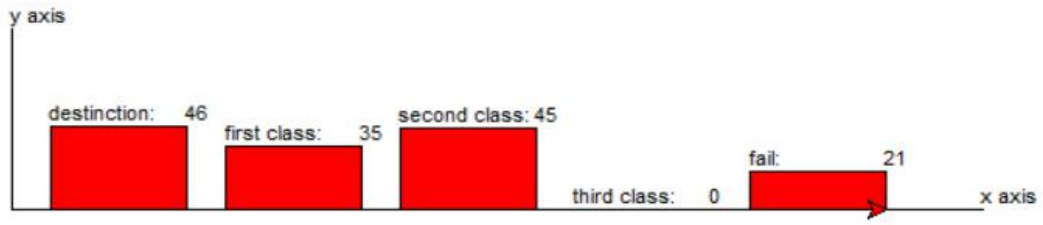
Source Code:

```
import turtle as t
destinction,first,second,third,fail=0,0,0,0,0
t.up()
t.goto(-300,0)
t.down()
t.lt(90)
t.fd(100)
t.write(str("y axis"))
t.bk(100)
t.rt(90)
t.fd(500)
t.write(str("x axis"))
t.bk(500)
def drawbar(height,marks):
    t.fd(20)
    t.fillcolor("red")
    t.begin_fill()
    t.left(90)
    t.forward(height)
    t.write(str(marks))
    t.right(90)
    t.forward(70)
    t.write(str(height))
    t.right(90)
    t.forward(height)
    t.left(90)
    t.end_fill()
a=[]
mark=["destinction:","first class:","second class:","third class: ","fail:"]
for line in open("python.txt","r"):
     b=int(line[5:])
     a.append(b)
print(a)
for i in a:
    if(i>=80):
        destinction+=1
    elif(i>=70):
        first+=1
    elif(i>=60):
        second+=1
    elif(i>=50):
        third+=1
```

```
    else:
        fail+=1
drawbar(destinction,mark[0])
drawbar(first,mark[1])
drawbar(second,mark[2])
drawbar(third,mark[3])
drawbar(fail,mark[4])
```

Output:

```
======== RESTART: D:/Basics of Python Programming/Lab Programs/graph.py ========
[80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 6
4, 80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 64, 80, 68, 64, 88, 88, 88, 88, 88,
88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88, 88
, 88, 88, 88, 88, 88, 88, 88, 88, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67, 67,
67, 67, 67, 67, 67, 67, 67, 67, 67, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15
, 15, 15, 15, 15, 15, 15, 15, 15, 15, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75,
75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75, 75
, 75, 75, 75]
```



22. Design a Python script to determine the difference in date for given two dates in YYYY:MM:DD format(0 <= YYYY <= 9999, 1 <= MM <= 12, 1 <= DD <= 31) following the leap year rules.

Source Code:

```
from datetime import datetime
day1 = input("enter the date in this format (yyyy:mm:dd)")
day2 = input("enter the date in this format (yyyy:mm:dd)")
day1 = datetime.strptime(day1, '%Y:%m:%d').date()
day2 = datetime.strptime(day2, '%Y:%m:%d').date()
diff = day2 - day1
print (diff.days)
```

Output:

```
In [8]: runfile('C:/Users/naray/.spyder-py3/temp.py', wdir='C:/Users/naray/.spyder-py3')

enter the date in this format (yyyy:mm:dd)2000:06:20

enter the date in this format (yyyy:mm:dd)2022:02:22
7917
```

23. Design a Python Script to determine the time difference between two given times in HH:MM:SS format.(0 <= HH <= 23, 0 <= MM <= 59, 0 <= SS <= 59)

Source Code:

```
def removecolons(s):
    if(len(s)==4):
        s=s[:1]+s[2:]
    if(len(s)==5):
        s=s[:2]+s[3:]
    return int(s)
def diff(s1,s2):
    time1=removecolons(s1)
    time2=removecolons(s2)
    hourdiff=time2//100-time1//100-1;
    minDiff=time2%100+(60-time1%100)
    if(minDiff>=60):
        hourdiff+=1
        minDiff=minDiff-60
    res=str(hourdiff)+':'+str(minDiff)
    return res
s1=input("enter the time in this format(HH:MM):")
s2=input("enter the time in this format(HH:MM):")
print(diff(s1,s2))
```

Output:

```
In [1]: runfile('C:/Users/naray/.spyder-py3/temp.py', wdir='C:/Users/naray/.spyder-py3')

enter the time in this format(HH:MM):07:00

enter the time in this format(HH:MM):12:00
5:0
```

Lab Outcomes:

Student should be able to

1. Design solutions to mathematical problems.
2. Organize the data for solving the problem.
3. Develop Python programs for numerical and text based problems.
4. Select appropriate programming construct for solving the problem.
5. Illustrate object oriented concepts.

Reference Books:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, "How to Think Like a Computer Scientist: Learning with Python 3", 3rd edition, Available at http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf
2. Paul Barry, "Head First Python a Brain Friendly Guide" 2nd Edition, O'Reilly, 2016
3. DainelY.Chen "Pandas for Everyone Python Data Analysis" Pearson Education, 2019

List of COs	PO no. and keyword	Competency Indicator	Performance Indicator
CO1	PO1: Engineering Knowledge	1.4	1.4.1
CO2	PO 2: Problem analysis	2.2	2.2.4
CO3	PO1: Engineering Knowledge	1.3	1.3.1
CO4	PO1: Engineering Knowledge	1.4	1.4.1