In [47]:
```
def createMatrix(rowCount, colCount, dataList): # function which converts data ir
    mat = []
    for i in range(rowCount):
        rowList = []
        for j in range(colCount):
            # you need to increment through dataList here, like this:
            rowList.append(dataList[rowCount * i + j])
        mat.append(rowList)

    return mat
with open('data1.txt') as A:
    contents1 = A.read()
    print(f"**********Data in first file ********** \n {contents1}")
with open('data2.txt') as B:
    contents2 = B.read()
    print(f"**********Data in second file ********** \n {contents2}")
with open('data3.txt') as C:
    contents3 = C.read()
    print(f"**********Data in third file ********** \n {contents3}")
with open('data4.txt') as D:
    contents4 = D.read()
    print(f"**********Data in fourth file ********** \n {contents4}")
```

```
**********Data in first file **********
 123456789

**********Data in second file **********
 980432128
**********Data in third file **********
 503
**********Data in fourth file **********
 707
```

In [48]:
```
matA = createMatrix(3,3,contents1)
print(f"Data in first file is converted to matrix as \n A = {matA}")
matB = createMatrix(3,3,contents2)
print(f"Data in second file is converted to matrix as \n B = {matB}")
matC = createMatrix(1,3,contents3)
print(f"Data in third file is converted to matrix \n C = {matC}")
matD = createMatrix(1,3,contents4)
print(f"Data in fourth file is converted to matrix \n D = {matD}")
```

```
Data in first file is converted to matrix as
 A = [['1', '2', '3'], ['4', '5', '6'], ['7', '8', '9']]
Data in second file is converted to matrix as
 B = [['9', '8', '0'], ['4', '3', '2'], ['1', '2', '8']]
Data in third file is converted to matrix
 C = [['5', '0', '3']]
Data in fourth file is converted to matrix
 D = [['7', '0', '7']]
```

In [141]:
```
# Multiplying matrices A and B
result = [[0, 0, 0, ],
        [0, 0, 0, ],
        [0, 0, 0, ]]

# iterating by row of A
for i in range(len(matA)):

    # iterating by column by B
    for j in range(len(matB[0])):

        # iterating by rows of B
        for k in range(len(matB)):
            result[i][j] += int(matA[i][k]) * int(matB[k][j])

print(f"A x B = {result}")

# Multiplying matrices C and D
result = [0]

# iterating by row of C
for i in range(len(matC[0])):

    result[0] += int(matC[0][i]) * int(matD[0][i])
print(f"C x D = {result}")

# Multiplying matrices A and C
result = [[0, 0, 0,]]

# iterating by row of C
for i in range(len(matC)):

    # iterating by column by A
    for j in range(len(matA[0])):

        # iterating by rows of A
        for k in range(len(matA)):
            result[i][j] += int(matC[i][k]) * int(matA[k][j])
print(f"C x A = {result}")


# Multiplying matrices B and D
result = [[0, 0, 0,]]

# iterating by row of D
for i in range(len(matD)):

    # iterating by column by B
    for j in range(len(matB[0])):

        # iterating by rows of B
        for k in range(len(matB)):
            result[i][j] += int(matD[i][k]) * int(matB[k][j])
print(f"B x D = {result}")
```

```
A x B = [[20, 20, 28], [62, 59, 58], [104, 98, 88]]
C x D = [56]
C x A = [[26, 34, 42]]
B x D = [[70, 70, 56]]
```

In [10]:
```
'''Ques 2 Define your own class / structure myComplex and calculate the sum, proc
and division of two complex numbers. Also calculate the conjugate, modulus
and phase angle of a complex number.'''



# Python3 program to
# demonstrate instantiating
# a class

#a,b,c,d=input("Enter Real part of first complex number"),input("Enter Imaginary
# Python3 program to show that we can create
# instance variables inside methods

# Class for Complex Number
class myComplex:

    def __init__(self, a, b, x,y ):
        # Instance Variable
        self.a = a
        self.b = b
        self.x = x
        self.y = y
        self.mod1 = (self.a**2+self.b**2)**(1/2)
        self.mod2 = (self.x**2+self.y**2)**(1/2)


    # Adds  complex numbers
    def Add(self):
        self.sum_real = self.a + self.x
        self.sum_imag = self.b + self.y
        return(print(f"Sum of complex numbers is {self.sum_real} + {self.sum_imag
    def Conjugate(self):
        return(print(f"Conjugate of {self.a} + {self.b}i is {self.a} - {self.b}i
    #product
    def Product(self):
        self.product_real = self.a*self.x - self.b*self.y
        self.product_imag = self.b*self.x + self.a*self.y
        return(print(f"Product of complex numbers is {self.product_real} + {self.
    # Retrieves instance variable
    def Modulus(self):
        return(print(f"Modulus of  {self.a} + {self.b}i is {self.mod1} \nModulus
    # division
    def Div(self):
        self.div_real = self.a*self.x + self.b*self.y
        self.div_imag = self.b*self.x - self.a*self.y
        modu = (self.x**2+self.y**2)
        return(print(f"Division of complex number  {self.a} + {self.b}i by  {self
    def phase(self):
        import math
        theta=math.atan(abs(self.b/self.a))
        if self.a>0 and self.b>0:
            return(print(f"Phase of complex number {self.a} + {self.b}i is :{thet
        elif self.a<0 and self.b>0:
            return(print(f"Phase of complex number {self.a} +{self.b}i is :{math.
        elif self.a>0 and self.b<0:
```

```
            return(print(f"Phase of complex number {self.a} + {self.b}i is :{-the
        elif self.a<0 and self.b<0:
            return(print(f"Phase of complex number {self.a}+ {self.b}i is :{theta
    def phase2(self):
        import math
        theta1=math.atan(abs(self.y/self.x))
        if self.x>0 and self.y>0:
            return(print(f"Phase of complex number {self.x} + {self.y}i is :{thet
        elif self.x<0 and self.y>0:
            return(print(f"Phase of complex number  {self.x} +{self.y}i is :{math
        elif self.x>0 and self.y<0:
            return(print(f"Phase of complex number  {self.x} + {self.y}i is :{-th
        elif self.x<0 and self.y<0:
            return(print(f"Phase of complex number {self.x }+ {self.y}i is :{thet


    # Note that the phase is calculated in radians


# Driver Code
object = myComplex(3,4,12,5)
object.Product()
object.Div()
object.Modulus()
object.Conjugate()
object.phase()
object.phase2()
```

```
Product of complex numbers is 16 + 63i
Division of complex number  3 + 4i by  12 + 5i  is 0.33136094674556216 + 0.1952
662721893491i
Modulus of  3 + 4i is 5.0
Modulus of  12 + 5i is 13.0
Conjugate of 3 + 4i is 3 - 4i
Conjugate of 12 + 5i is 12 - 5i
Phase of complex number 3 + 4i is :0.9272952180016122
Phase of complex number 12 + 5i is :0.39479111969976155
```

In [5]:
```
# Ques3 Find the average distance between two points on a straight line made of N
#points.

def find_avg(N):

    distance = 0
    for i in range(N):  # Selecting one point in line
        for j in range(N):# Selecting any other point other than the previously s
            distance = distance + abs(j-i)
    return(f"Avg of distance of points with {line} points in line is {distance/li
find_avg(line)
```

Out[5]: 'Avg of distance of points with 2 points in line is 0.5 units '

In [6]:
```
line = int(input("Enter the Number of points in the line"))
find_avg(line)
```

Out[6]:
```
'Avg of distance of points with 3 points in line is 0.8888888888888888 units '
```

In [7]:
```
line = int(input("Enter the Number of points in the line"))
find_avg(line)
```

Out[7]:
```
'Avg of distance of points with 4 points in line is 1.25 units '
```

In [1]:
```
# Ques 4

#Designing a Hangman game

#This is a letter search algorithm which returns the indices that a particular le
def letter_search(letter,string):
    indices=[]
    for i in range(len(string)):
        if string[i].lower() == letter.lower():
            indices.append(i)
    return(indices)
```

In [5]:
```
def play_Hangman():
    #The Names that are to be guessed by the player
    Names=['Belgium', 'Brazil' , 'France' , 'England' , 'Italy' , 'Argentina' , '
    import random
    n=random.randint(0,23)
    #Number of wrong guesses is 40% of the word length
    guess_lft=int(len(Names[n])*2/5)
    print("Welcome to Hangman: 2021")
    unknown_word="?"*len(Names[n])#Unknown name
    print("Your word is: "+unknown_word)
    print("You can make at most "+str(guess_lft)+" wrong guesses")
    print("Please enter only one character at a time")
    while True:
        let=input('Guess a letter of the name: ')
        if len(let)==1:#player can enter only one character at a time
            indices=letter_search(let,Names[n])
            if len(indices)>0:#correct guess
                for i in indices:
                    unknown_word=unknown_word[:i]+let+unknown_word[i+1:] # reveal
                print("Correct guess! The word is: "+unknown_word+"\t\twrong gues
                if len(letter_search('?',unknown_word)) ==0:
                    print("Congratulations, You won!")#Nothing left to reveal, th
                    break
            else:#wrong guess
                guess_lft=guess_lft-1
                if guess_lft<0:
                    print("You lost! the correct word is "+Names[n])#no wrong gue
                    break
                print(let+" is a wrong guess! The word is: "+unknown_word+"\t\t\t
        else:
            print("Enter only a single letter!")
```

In [6]:
```
play_Hangman()
```

```
Welcome to Hangman: 2021
Your word is: ?????
You can make at most 2 wrong guesses
Please enter only one character at a time
g is a wrong guess! The word is: ?????                wrong guesses left: 1
a is a wrong guess! The word is: ?????                wrong guesses left: 0
You lost! the correct word is Chile
```

In [10]:
```
play_Hangman()
```

```
Welcome to Hangman: 2021
Your word is: ???????
You can make at most 2 wrong guesses
Please enter only one character at a time
Correct guess! The word is: ????a??            wrong guesses left: 2
d is a wrong guess! The word is: ????a??                  wrong guesses l
eft: 1
f is a wrong guess! The word is: ????a??                  wrong guesses l
eft: 0
You lost! the correct word is Germany
```

In [12]:
```
play_Hangman()
```

```
Welcome to Hangman: 2021
Your word is: ?????????
You can make at most 3 wrong guesses
Please enter only one character at a time
Correct guess! The word is: a???????a          wrong guesses left: 3
Correct guess! The word is: ar??????a          wrong guesses left: 3
Correct guess! The word is: arg?????a          wrong guesses left: 3
Correct guess! The word is: arge????a          wrong guesses left: 3
Correct guess! The word is: argen??na          wrong guesses left: 3
Correct guess! The word is: argent?na          wrong guesses left: 3
Correct guess! The word is: argentina          wrong guesses left: 3
Congratulations, You won!
```

In [ ]:

In [ ]: