INTRODUCTION OF PYTHON

1)What is python?

2)Who invented python?

3)What are the features of python?

4)Who uses python?

5)How to download and install python?

1)Python:

A programming language which can be used almost in most of the developmet, creation etc..

Ex:web development,application designing, graphical designing.

Pyhton is an interpreter(line by line execution) where as C is a compiler.

2)Invention:

In 1980's **JUDO VAN ROSSUM** started creating a programming launguage which is very simple to understand and beautiful to look and it uses significant indentations(:)

instead of braces to distinguish what block of code comes under which class or function.

(Braces: A syntactic construct in programming launguage like we use { } in C/C++.)

The name was inspired from MONTY PYTHON a cartoon show creator and released in 1989.Pyhton got popularity with the rise of Artificial Intellilgence and Machine Learning(AI&ML).

3)Features:

Python is easy to write, easy to understand also very efficient and it is a high level, Interpreted language(A programming language which doesn't need earlier compiling) with easy syntax and dynamic semantics(The message conveyed by words,sentences and symbols) or (Don't need to initialize anything).

By the line-by-line execution it will be easy to debugging errors much more easier.

(Debugging is a process of solving errors in a program. This is a reason for the significant slow procesing of code)

#Simplicity(easy to use)

#Open source(free to use & available around the globe)

#Portable(python code can be written in one computer and executed in another computer without any errors)

#Embeddable and Extensible(python allows other programming launguages like C/C++ to be embedded into it)

#Interpretation(Doesn't need more memory as it executes line by line.Line by line execution doesn't take memory and usage of CPU as a whole programme takes.

That means both CPU and Memory management are handled by python itself)

#Huge library(python have a huge set of libraries such as Numpy, Matplotlib and Scikit-learn that are useful in problem solving)

#Object orientation(You can give a real world problem into code and help provide security to it)

4)Who use python:

Thousands of companies use python for their daily business standards like

1)Google

Google uses python to provide best search results

2)Dropbox

Data storing software's servers are also maintained by using python

3)NSA(National Security Agency)

For cyber security analysis and aslo for encryption and decryption of data

4)Bit torrent

To share files to users through the servers

5)RaspberyPi(it is an OS)

coded with python

6)NASA

Scientists uses python to do scientific calculations

7)Netflix

Netflix uses ML to recommend TV shows according to users past activity

8)Youtube

For better search recommendations and search results

5)How to download and install python?

To do work using python we need to dowload and install it from official python website "python.org".

While in the beginning of the installation must check the box "Add python to path" and then click install now and after that

click on "disable the path length".

After the installation search for "IDLE" in windows search and open it. There you can start using python.

~To open a new file:1)Ctrl+N or

2)File->New

~After opening a new file save it earlier.To save the

file:1)Ctrl+S or

2)File->Save

~A python file must be saved with the extension of ".py"

~To execute the code press function key "F5" or

Run -> Run Module

~To open an existed file:1)Ctrl+O or

2)File->Open

Python can be run in other idles from different providers like Pycharm,Jupiter(notebook).

PYTHON PROGRAMMING

1)Comments:

A python comment is a line of text that appears in code but is not executed by the program.

A comment can be declared by using hashtag(#).

Python comments are used to explain how code works and for testing purpose.

Ex: **#**comments are used to explain the program

2)Keywords:

Keywords are the reserved words in python. We cannot use keywords as name,function,variable or any other identifier.

They are used to define the syntax and structure of pyhton.

The keywords are case sencitive.

There are 35 keywords in python 3.10.5 (july-2022).

Ex: print,len,as,in,and,or,break,def,del,class,lambda,if,else,elif

etc….

3)Operators:

Symbols that are used to perform arithmetic and logical computation.

1)Arithmetic operators(+,-,*,/,%)

2)Assigning operators(=)

4)Printing:

To print the data we use the keyword **print**

printing a word/number/sentence

1)**print**("123") #data within " " or ' ' will be printed as it is

2)**print**("hello welcome to python")

printing a word multiple times in single line without space

print("hello"*10)

printing a word multiple times in single line with space

print("hello "*10)   #leave space after the word

printing a word multiple times with long space

print("hello\t"*5)   # \t is used to print with long space

printing a word multiple times in new line

print("hello\n"*10)   # \n is used to print in new line

5)Variable:

A variable is a storage container that contains the assigned value or data given by the user.

a=2   #assigning value 2 to a using assigning operator

print(a) #printing the assigned value.

#Here we don't have ' ' or " ". Then in the absence of ' ' or " " the value of the variable will be printed not variable.

6)DataType:

The classification or categorization of knowledge items is called the datatype.

1)Numeric:1)int(numbers from 0 to infinity)

2)float(decimal numbers like 1.4,7.98,0.125)

3)complex numbers(a+ib,i– imaginary term)

2)Sequence:1)strings (collection of characters)

2)lists (ordered,indexed,mutable)

3)tuple(orderd,indexed,immutable,stores)

multiple datatypes,can store duplicate values)

3)Boolean: Based on True or False.

1)If true – the value is 1

2)If false – the value is 0

4)Array (Data structure that holds a collection of

elements having the same data types)

5)Set (unordered,elements are unique,no duplicate

elements,set itlesf may be modified but element of

the set are immutable)

6)Dictionary(unordered, **key-value** pair,keys are

unique and immutable.)

7)Flow control:

The order in which a program's code executes.

The flow control is regulated by using 'conditional statements' such as loops and function calls(for,while).

Types of flow controls:1)Sequential:Default

2)Selection:used for decision making (using 'if')

3)Repetition:used for looping(using 'for','while')

8)Taking input from the user:

In previous case, while printing a value/data you have already assigned to the variable. Now we are going to store the input from the user in the variable.

#To take input from the user we use the keyword **input**

#By default python takes the input in the form of string

a=**input**()

print(a)

while running the above code, you won't have any clear idea about what to enter there. So to get a clear idea to enter the correct data in the output screen we are going to print the info required for getting the input.

As we know that the data within ' ' and " " will be printed as it is. We are going to use the same as shown below

a=**input**("enter any number/word: ")

print(a)

For more clearance about the data that what you actually gave the input you can write in print() as shown below

a=**input**("Enter any number/word: ")

print("The data you entered is: ",a)

Here in the print The data you entered is: will be printed as it is, as it is located within " ". Then we used comma(,) for separating the text and the data assigned to the variable a by the user by **input** keyword. You can observe that a is not within ' 'or " " so then the value/data given to a will be printed.

9)Arthematic Operations:

    1)Addition (+)   2)Substraction (-)   3)Multiplication (*)

           4)Division (/)   5)Modulus (%)

a=23		a=23
b=32	(or)	b=32
print(a+b)		c=a+b #3rd variable
print("sum of a,b is: ",a+b)		print("sum of a,b is: ",c)

Input method:

#Converting one datatype into another datatype is called
  as **TypeCasting**.

a=**int**(input("enter a:")) #Converting string to integer form

b=**int**(input("enter b:"))

print("sum of a,b is:",a+b)

10) Strings:

String is a collection of characters.Strings are immutable data type in python. We use single quotes(' ') or double quotes(" ") to declare the strings. Strings are case sensitive.

name='python programming'

print(name)

(or)

name="python programming"

print(name)

To access the value in a string we need to use the **indexing** method. from left In any programming the index value starts from 0 (zero) side and also the negative index values starts from -1 from right side.

Lets take a string "PYTHON"

P	Y	T	H	O	N
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

If there is any space between two words then the space also considered as an element in strings.

Lets take a string "PYTHON PROGRAM"

P	Y	T	H	O	N		P	R	O	G	R	A	M
0	1	2	3	4	5	6	7	8	9	10	11	12	13
-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

To access an element in a string we have to specify the index value of the particular element.

```
a="python programming"
1)print(a)
2)print(a[3])
3)print(a[:])    # slicing
4)print(a[:8])  # the last element won't be printed (n-1)
5)print(a[4:])
6)print(a[2:9])
7)print(a[1:13:2])     #[start:stop:step]  (n-1)
8)print(a[::-1]) #reverse of string
9)print(a[13:0:-3])
```

11)String Methods:

```
# To change the starting letter into capital letter
    a="python"
    print(a.capitalize()) # capitalize method
# To convert all the small letters into capital letters
    a="python"
    print(a.upper()) # upper method
# To convert all the capital letters into small letters
    a="python"
    print(a.lower()) # lower method
```

#To replace an element with another element

a="python"

print(a.replace('y','$'))

To count the number of occurances of an element

a="programming language"

print(a.count('r'))

#To split a long string into two strings

a="python is a programming language"

print(a.split("is"))

To find the index number of an element

a="programming language"

print(a.find('i'))

If there are repeatative elements in a string, the find method gives the first element's index number.

a="programming language"

print(a.find('a'))

We can also use the **index**() method to find the index number of an element works same as find() method.

a="programming language"

print(a.**index**('a'))

12)String elements and types

To find the type of elements in a string

(alphabets,numbers,lowercase,uppercase,alphanumerical etc..)

If the condition is true then the output will be as True, otherwise the output will be as False.

```python
1)a="python"
   print(a.isalpha())   #checking alphabets
2)a="12345"
   print(a.isnumeric())      #checking numbers
3)a="python123"
   print(a.isalnum())   #checking combination of alphabets
                              and numbers
4)a="PYTHON"
   print(a.isupper())   #checking uppercase letters
5)a="python"
   print(a.islower())   #checking lowercase letters
```

13)String Operations

1)String Concatination: Adding two strings as single string

```python
a="python"
b="programming"
print(a+b)
```

2)Percentage format (%s)

```python
a="bus"
b="home"
c="if you catch the %s, you will reach your %s"%(a,b)
#a takes place at 1st %s, b takes place at 2nd %s
print(c)
```

3)Bracket format ({ })

```python
a="bread"
```

b="butter"

breakfast="todat my breakfast is {} and {}".format(a,b)

(or)

breakfast=f"today my breakfast is {a} and {b}"

print(breakfast)

14)Lists:

Lists are ordered and contains multiple data types which are mutable(changable).Each element can be accessible using index values.

To create a list use [ ] or use the keyword **list**()

mylist=[1,4.65,'words',17+9j]

print(mylist)

(or)

mylist=**list**(1,4.65,"words",17+9j)

print(mylist)

15)Accessing elements from a list:

mylist=[1,2.54,'words','hi welcome to python',2-8j,'python']

1)print(mylist[:])   #slicing

2)print(mylist[3])

3)print(mylist[0:2])

```python
4)print(mylist[3:5])

5)print(mylist[2:])

6)print(mylist[::1])

7)print(mylist[::-1])

8)print(mylist[1:5:2])

9)print(mylist[3][6])


#Length of a list:
    mylist=[1,2,3,4,5,6]
    print(len(mylist))
#add passed elements as single element
    mylist=[1,5.3,'asdf','pyhton',16-2j]
    mylist.append([12,'hello'])
    print(mylist)
#extending list by adding elements one by one
    mylist=[1,5.3,'asdf','pyhton',16-2j]
    mylist.extend([12,'hello'])
    print(mylist)
#inserting an element in a specific place using index
    mylist=[1,5.3,'asdf','pyhton',16-2j]
    mylist.insert(2,'hello')
    print(mylist)
#list concatination
    mylist=[1,2,3,4,5]
    print(mylist + ['numbers'])
```

```
#multiple list elements
    mylist=[1,2,3,4,5]
    print(mylist*2)
#deleting elements from list using del keyword & index value
    a=[1,2,3,4,5,6,7,8]
    del a[3]
    print(a)
#removing element from list using remove() method
    a=[1,2,3,4,5,6,7,8]
    a.remove(3)
    print(a)
#poping elements in a list using pop() method using index
    a=[1,2,3,4,5,6,7,8]
    b=a.pop(4)      #pop returns to the poped element
    print("poped element: ",b)
    print("remaining list: ",a)
    if there is no argument in pop() then the last element will
be poped from the list
    a=[1,2,3,4,5,6,7,8]
    a.pop()
    print(a)
#clearing the elements in a list using clear() method
    a=[a,s,d,f,g,h]
    a.clear()   #clears the elements in list not the list
    print(a)
```

```
#deleting a list using del() keyword
    a=[a,s,d,f,g,h]
    del [a]
    print(a) #shows an error
#counting number of repetations of an element in a list
    a=[1,2,3,2,5,1,4,2]
    print(a.count(2))
#sorting elements in assending order
    a=[23,1,56,0,3,55,10,7,34,6]
    a.sort()  # sorts the list completely
    print(a)
        (or)
    a=[23,1,56,0,3,55,10,7,34,6]
    print(sorted(a)) #sorts the elements in the list but not
                        change the list
    print(a)


#printing the elements in dessending order
    a=[23,1,56,0,3,55,10,7,34,6]
    a.sort(reverse=True)
    print(a)
#printing the list elements in reverse order
    a=[1,2,3,4,5,6,7]
    a.reverse()
    print(a)
```

#copying a list into another new empty list using **copy**() method

```
a=[1,2,3,4,5,6,7]
b=[]
b=a.copy() #copies the elements of list to the new list
print(b)
```

16)Tuples:

Tuples are as lists with one exception, that they are immutable.Also tuples are faster than lists. Tuples can be created using ( ) and/or **tuple**() keyword.

```
t=(1,2,3,4,5,"python")
print(t)
        (or)
t=tuple((1,2,3,4,5,"python"))
print(t)
```

#Tuple concatination (**+**)

```
my_tuple=()
my_tuple=my_tuple+(1,2,3,4)
print(my_tuple)
```

#creating a tuple with single element using comma(**,**)

```
a="tuple", #added comma(,) after the string to create tuple
print(a)
```

17)Acessing elements of Tuples

#printing the elements in the tuple

```
t=(1,2,3,4,5,6,7)
print(t[:])
print(t[3])
print(t[1:5])
print(t[::1])
```

#changing the elements in a tuple if there is any mutable data type in the tuple

```
t=(1,2,3,4,5,["hi","python",75],6,7)
t[5][2]="programming"
print(t)
```

#counting the number of repetations of elements with **count**() method

```
a=(1,4,2,3,1,4,2,3,4,4,1,2,3,4,3,2,1)
print(a.count(3))
```

#finding the index value of an element using **index**() method

```
a=(1,2,3,4,5)
print(a.index(3))
```

18)Boolean:

The boolean represents one of two values, either **True** or **False**.

The comparision operators such as =,!=,>,<,>=,<= are used to compare two values.If the comparision is correct, then the output is True.If not the output is False.

The logical operators such as and,or,not are used on booleans.

#Using comparison operators

number=5

1)print(number **<** 10) #if the condition is correct then the

output is True, else output is False.

2)print(number **>** 0)

3)print(number **==** 5) #equal

4)print(number **!=** 0) #not equal

5)print(number **<=** 5)

6)print(number **>=** 5)

#Using logical operators

age=18

gender="male"

1)print(age==18 **and** gender=="male")

2)print(age>=18 **or** gender=="male")

3)print(**not** gender=="male")

19)Arrays

An array is a data structure which can hold more than one value at a time.It is a collection or ordered series of elements of the same data types.That means an array takes only a single data type elements.

To create an array we have to **import array module**.

20)Accessing elements from arrays:

Array elements are indexe and mutable.So we can easily add/change and remove/delete elements from the array.

#We can find the length of an array using **len** keyword

#Array concatination also possible using arthematic operator(**+**)

#We can perform slicing and looping also

21)Sets:

Sets are un-ordered collection of unique elements.They are mutable. To create a set we have to use flower brackets**{ }**.

#creating a set

my_set={1,2,3,4}

print(my_set)

#creating a set with duplicate elements

my_set={1,2,3,4,4,4,4,4}

print(my_set) # since the elements in set are unique, the

duplicate elements will be printed only

once

22)Accessing elements in a set:

#adding an elements in set

    a={1,3,5} #to add an element in a set, the set must

                contain elements

    a.add(7)

    print(a)

#operations in sets

    set1={1,2,3,4}

    set2={3,4,5,6}

    print(set1.**union**(set2)) #union(**|**)- adding the elements of

                    both sets and prints common elements once

    print(set1.**intersection**(set2)) #intersection(**&**)-prints the

                    commom elements

    print(set1.**difference**(set2)) #difference(**-**)-deletes the

                    common elements and prints

                    the remaining elements

    print(set1.symmetric_difference(set2))

                  #symmetric_difference(**^**)-deletes the common elements and prints the remaining elements

23)Logical operations in sets:

#Logical operations returns the boolean values(True or False)

set1={0,1,2,3,4,5,6,7,8,9}

set2={0,2,4,6,8}

set3={1,3,5,7,9}

1)print(set2**==**set3) #equal (or) symmetric

2)print(set2**!=**set3) #not equal (or) not symmetric

3)print(set2**<=**set1) #set2 is subset of set1

4)print(set2**>=**set3) #set3 is subset of set2

5)print(set2**<**set1) #set2 is a proper subset of set1

6)print(set2**>**set1) #set1 is a proper subset of set2

7)print(set3**<**set1) #set3 is a proper subset of set1

8)print(set3**>**set1) #set1 is a proper subset of set3

24)Dictionary:

A dictionary is a data type that holds a **key:value** pair.

These are mutable. To create a dictionary we have to use flower brackets **{ }** or we can use **dict**() keyword.

my_dict={ }

print(my_dict)

(or)

my_dict=dict()

print(my_dict)

#dictionary with key:value pairs

my_dict={"name":"python","work":"program",1:"code"}

print(my_dict)

(or)

```
my_dict=dict({"name":"python","work":"program",1:"code"})
print(my_dict)
```

25)Accessing elements from a dictionary:

#printing value by using key

```
d={1:"python",2:"java",3:"c",4:"c++"}
1)print(d[1])
2)print(d.get(2)) #using get() method
```

changing the elements in dictionary

```
#when the keys are integers
1)d={1:"python",2:"java"}
   d[1]=5  #changing with an integer value
   print(d)


2)d={1:"python",2:"java"}
   d[1]="coding" #changing with a string
   print(d)


#when the keys are strings
1)d={"first":"python","second":"java"}
   d["first"]=5  #changing with an integer value
   print(d)
```

```python
2)d={"first":"python","second":"java"}
    d[1]="coding"  #changing with a string
    print(d)
#adding elements in dictionary
    dictionary1={1:"python",2:"program"}
    dictionary1[3]="coding"
    print(dictionary1)
#deleting elements from dictionary using key
  1) d={1:"python",2:"program",3:"coding"}
    a=d.pop(3)  #poping value with key
    print("poped value: ",a)
    print("remaining dictionary: ",d)


  2) d={1:"python",2:"program",3:"coding"}
    a=d.popitem() #poping last key:value pair & return
                        as tuple
    print("poped key:value pair is: ",a)
    print("remaining dictionary is: ",d)
#clearing the dictionary using clear() method
    my_dict={1:"a",2:"b",3:"c",4:"d"}
    my_dict.clear()
    print(my_dict)
#getting all the keys from the dictionary
    my_dict={1:"a",2:"b",3:"c",4:"d"}
    print(my_dict.keys())
```

```python
#getting all the values from the dictionary
    my_dict={1:"a",2:"b",3:"c",4:"d"}
    print(my_dict.values())
#getting all the items from the dictionary
    my_dict={1:"a",2:"b",3:"c",4:"d"}
    print(my_dict.items())
```

PYTHON PROGRAMMS

Programming using python:

#Printing Hello world

```
print("hello world")
```

#Adding 2 numbers

```
a=2
b=4
c=a+b
print(c)
```

#Adding 2 numbers with user input

```
a=int(input("enter value of a: "))
b=int(input("enter value of b "))
print("sum of a,b is= ",a+b)
```

#Finding the square root,square,cube etc…. of a number

```
a=4
1)print(a**0.5)
2)print(a**2)
3)print(a**3)
4)print(a**4)
```

```python
#Finding the area of a triangle
    s1=float(input("enter side s1:"))
    s2=float(input("enter side s2:"))
    s3=float(input("enter side s3:"))
    s=(s1+s2+s3)/3
    area=(s*(s-s1)*(s-s2)*(s-s3))**0.5
    print("the are of triangle=",area)


#Solving a quadratic equation using cmath module
    #QE: a*x^2+b*x+c=0
    import cmath
    a=1
    b=2
    c=3
    d=(b^2-4*a*c)
    solution1=(-b+cmath.sqrt(d))/2*a
    solution2=(-b-cmath.sqrt(d))/2 *a
    print("the solutions are {} & {}".format(solution1,solution2))


#Swapping 2 variables using 3rd variable
    x=1
    y=2
    temp=x
    x=y
    y=temp
```

```python
        print(x,y)
#Swapping 2 variables without using 3rd variable
        x=1
        y=2
        x,y=y,x
        print(x,y)


#Printing a random integer in given range
        import random
        print(random.randint(0,9))


#Converting Celcius to Fahrenheit
        c=37.5
        f=(c*1.8)+32
        print(c,"degrees celcius is equal to",f,"degrees fahrenheit")


#Checking if a number is positive or negative
        a=float(input("enter a number to check: ")
        if a>0:
                print("number is positive")
        elif a<0:
                print("number is negative")
        else:
                print("number is zero")
```

```python
#Checking if a number is even or odd
    num=float(input("enter a number to check: ")
    if num%2==0:
        print(num,"is even")
    else:
        print(num,"is odd")


#Checking leap year
    year=int(input("enter a year"))
    if year%400==0 and year%100==0:
        print(year,"is a leap year)
    elif year%4==0 and year%100!=0:
        print(year,"is a leap year")
    else:
        print(year,"is not a leap year")


#Finding the largest among the three
    num1=5
    num2=7
    num3=9
    if num1>=num2 and num1>=num3:
        largest=num1
    elif num2>=num1 and num2>=num3:
        largest=num2
    else:
```

```
            largest=num3
    print("the largest number among three is: ",largest)



#Checking wheather a number is prime or not.
PrimeNumber: A positive integer greater than 1 which has no
factors other than 1 and the number itself.


    number=67
    if number>1:
            for i in range(2,num):
                    if num%i==0:
                            print(number,"is not prime")
                    else:
                            print(number,"is prime")
    else:
            print(number,"is not prime")



#printing all the prime numbers in given range
    initial=0
    final=50
    print("the prime numbers in the given range are:")
    for numbers in range(initial,final+1):
            if number>1:
```

```python
        for i in range(2,number):
            if(number%i==0):
                break
            else:
                print(number)




#Finding the factorial of a number
    n=int(input("enter a number"))
    factorial=1
    if n<0:
        print(n,"factorial does not exist for -ve numbers")
    elif n==0:
        print("factorial of zero is 1")
    else:
        for I in range(1,n+1):
            factorial=factorial*i
    print("factorail of",n,"=",factorial)




#Printing multiplication table of a number
    n=int(input("enter a number: "))
    for i in range(1,n+1):
        print(n,"x",I,"=",n*i)
```

```python
#Generating fibonacci sequence
    nterm=int(input("enter no.of terms: ")
    n1,n2=0,1
    if nterm<=0:
        print("please enter a +ve number")
    elif nterm==1:
        print("fibonacci sequence upto 1 is",n1)
    else:
        while count<nterm:
            print(n1)
            a=n1+n2
            n1=n2
            n2=a
            count=count+1


#Finding the area of a circle
    r=float(input("enter radius of circle: "))
    area=3.14*(r**2)
    print("area of circle is:",area)


#Checking armstrong number
153=1^3+5^3+3^3 is an armstrong number
    n=int(input("enter a number:"))
    sum=0
    temp=n
```

```python
        while temp>0:
            r=temp%10
            sum=sum+r**3
            temp=temp//10
        if n==sum:
            print(a,"is armstrong")
        else:
            print(a,"is not armstrong")


#printing sum of numbers in a range
    r=int(input("enter range:"))
    sum=0
    for i in range(0,r+1):
        sum=sum+i
    print(sum)


            (or)


    num=int(input("enter range:"))
    sum=0
    if num<=0:
        print("please enter a positive integer")
    else:
        while num>0:
            sum=sum+num
```

```python
        num-=1          #num=num-1
    print("the sum is",sum)


#finding the numbers that are divisible by another number
    n=int(input("enter a number to print divisible numbers:"))
    for i in range(0,100):
        if i%n==0:
            print(i)


#finding the ASCII(American Standard Code for Information
Interchange) of a character
    character=input("enter any character:")
    print("the ascii value of",character,"is:",ord(character))


#finding the factors of a number
    n=int(input("enter a number:"))
    for i in range(0,n+1):
        if n%i==0:
            print(i)


#addition of 2 matrices
    m=[[1,2,3],[4,5,6],[7,8,9]]
    n=[[1,2,3],[4,5,6],[7,8,9]]
    result=[[0,0,0],[0,0,0],[0,0,0]]
    for i in range(len(m)):
```

```python
        for j in range(len(n)):
                result[i][j]=m[i][j]+n[i][j]
    for r in result:
        print(r)


#checking string palindrome
    a="asdfdsa"
    b=a[::-1]
    if a==b:
        print("palindrome")
    else:
        print("not palindrome")
#sorting a string
    s=input("enter a word")
    st=" "
    for i in s:
        st=st+i
    f=sorted(st)
    print(f)


#counting no.of vowels in a string
    vowels="aeiou"
    string="hello welcome to python programming"
    count={}.fromkeys(vowels,0)
    for i in string:
```

```python
            if i in vowels:
                    count[i]+=1
        print(count)


#Accessing index value of elements in a list
        1)l=[1,2,3,4,5,6,7]
            for index, value in enumerate(l):
                    print(index,value)
        2)l=[1,2,3,4,5,6,7]
            for index, value in enumerate(l,start=1):
                    print(index,value)


#Printing multiple list elements in a single list/
  Flattening anested list
        1)l=[1,2,[3,4],5,6,7,[8,9,10]]
            flat_list=[num for sublist in l for num in sublist]
            print(flat_list)
        2)l=[1,2,[3,4],5,6,7,[8,9,10]]
            flat_list=sum(l,[])
            print(flat_list)


#Iterating over dictionaries using for loop
        a={1:'a',2:'b',3:'c',4:'d',5:'e',6:'f',7:'g'}
        for key,value in a.items:
                print(key,value)
```

```python
#Checking if a key is already present in a dictiionary
d={1:'a',2:'b',3:'c',4:'d',5:'e',6:'f'}
k=int(input("enter a key:"))
if k in d:
    print("key already in dictionary")
else:
print("key not in dictionary")


#finding the data type of an element
a=2
print(type(a))


#printing two lists into a dictionary
l1=[1,2,3,4,5]
l2=[a,b,c,d,e]
d=dict(zip(l1,l2))
print(d)


#counting the no.of digits in an integer
i=325916
count=0
while I !=0:
```

```python
    i=i//10

    count=count+1

print("no.of didgits=",count)
```