

USN:19BTRCY069

Syed Mohammad Wazidul Ifkat

LAB PROGRAM 1

1(A)To elaborate variables and their data Types such as int,float,boolean,string,list,set,dictionary and tuples

```
In [1]: a = 10
b = 4.0
d = True
c ="Python for Data Science"
e = [4, 3.0, "V Semester", "Harry"]
f = {4, 4, 2, 1, 2, 5, 7, 4}
g = (4,3.0, "Python mania", "Harry")
h = {"a" :4.5, "Python": "Language"}

print("Value of a=",a)
print("Type of a=",type(a))

print("Value of b=",b)
print("Type of b=",type(b))

print("Value of c=",c)
print("Type of c=",type(c))

print("Value of d=",d)
print("Type of d=",type(d))

print("Value of e=",e)
print("Type of e=",type(e))

print("Value of f=",f)
print("Type of f=",type(f))

print("Value of g=",g)
print("Type of g=",type(g))

print("Value of h=",h)
print("Type of h=",type(h))
```

Value of a= 10
Type of a= <class 'int'>
Value of b= 4.0
Type of b= <class 'float'>
Value of c= Python for Data Science
Type of c= <class 'str'>
Value of d= True
Type of d= <class 'bool'>
Value of e= [4, 3.0, 'V Semester', 'Harry']
Type of e= <class 'list'>
Value of f= {1, 2, 4, 5, 7}
Type of f= <class 'set'>
Value of g= (4, 3.0, 'Python mania', 'Harry')
Type of g= <class 'tuple'>
Value of h= {'a': 4.5, 'Python': 'Language'}
Type of h= <class 'dict'>

Exchange of Two values(int,float,string)

```
In [2]: a=int(input("Enter the first number"))
b=int(input("Enter the second number"))
print("Before swapping a=",a,"and b=",b)
a,b=b,a
print("After swapping a=",a,"and b=",b)
```

Enter the first number30
Enter the second number40
Before swapping a= 30 and b= 40
After swapping a= 40 and b= 30

```
In [3]: a=float(input("Enter the first number"))
b=float(input("Enter the second number"))
print("Before swapping a=",a,"and b=",b)
a,b=b,a
print("After swapping a=",a,"and b=",b)
```

Enter the first number3.5
Enter the second number4.5
Before swapping a= 3.5 and b= 4.5
After swapping a= 4.5 and b= 3.5

```
In [4]: a=input("Enter the first string")
b=input("Enter the second string")
print("Before swapping a=",a,"and b=",b)
a,b=b,a
print("After swapping a=",a,"and b=",b)
```

Enter the first stringSyed
Enter the second stringIfkat
Before swapping a= Syed and b= Ifkat
After swapping a= Ifkat and b= Syed

```
In [ ]:
```

1(B)To elaborate mathematical operations such as addition,substraction,multiplication,division,modulo and power

```
In [5]: a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
sum=a+b
diff=a-b
prod=a*b
div=a/b
rem=a%b
pow=a**b
flr=a//b
print("sum of {0} and {1} is : {2}".format(a,b,sum))
print("Difference of {0} and {1} is : {2}".format(a,b,diff))
print("Product of {0} and {1} is : {2}".format(a,b,prod))
print("{0} divided by {1} is : {2}".format(a,b,div))
print("{0} remainder {1} is : {2}".format(a,b,rem))
print("{0} power {1} is : {2}".format(a,b,pow))
print("{0} floor division {1} is : {2}".format(a,b,flr))
```

Enter the first number:50
Enter the second number:25
sum of 50 and 25 is : 75
Difference of 50 and 25 is : 25
Product of 50 and 25 is : 1250
50 divided by 25 is : 2.0
50 remainder 25 is : 0
50 power 25 is : 2980232238769531250000000000000000000000
50 floor division 25 is : 2

Explore the operator precedence

```
In [6]: a = 20
b = 10
c = 10
d = 5
e = (a + b) * c / d #( 30 * 10 )/5
print("Value of (a + b) cd is ",e)
e = ((a + b) * c) / d # (30 * 10 ) / 5
print("Value of ((a + b) c) / d is ",e)
e = (a + b) * (c/d) # (30) * (10/5)
print("Value of (a + b) (c/ d) is ",e)
e = a + (b* c) / d # 20 + (100/5)

c = 10

print("value of a + (b* c) / d is ",e)
```

Value of (a + b) cd is 60.0
Value of ((a + b) c) / d is 60.0
Value of (a + b) (c/ d) is 60.0
value of a + (b* c) / d is 40.0

```
In [ ]:
```

Name:Syed Mohammad Wazidul Ifkat

USN:19BTRCY069

Lab Program 2

2(A)Python program to find the sum and average of natural numbers up to n where n is provided by user

```
In [2]: num = int(input("Enter the number of terms: "))
num1 = num
if num < 0:
    print("Please enter a positive number")
else:
    sum = 0
    while(num > 0):
        sum += num
        num -= 1
    print("The sum of first {} natural numbers is:{}".format(num1,sum))
    print("The average of first {} natural numbers is:{}".format(num1,sum/num1))
```

Enter the number of terms: 10
The sum of first 10 natural numbers is:55
The average of first 10 natural numbers is:5.5

2(B)Python program to find Factorial,and Fibonacci of a number,received by user,with iterative as well as recursive process

```
In [6]: #To find the factorial of a number(iterative method)
def fact_iter(n):
    f=1
    i=1
    while(i<=n):
        f=f*i
        i=i+1
    return f

n=int(input("Enter the number:"))
if(n<0):
    print("Invalid input")
elif(n==0):
    print("Factorial of 0 is 1")
else:
    f=fact_iter(n)
    print("Factorial of {} is: {}".format(n,f))
```

Enter the number:5
Factorial of 5 is: 120

```
In [12]: #To find the factorial of a number(Recursive method)
def fact_rec(n):
    if(n==0 or n==1):
        return 1
    else:
        return(n*fact_rec(n-1))

n=int(input("Enter the number:"))
if(n<0):
    print("Invalid input")
else:
    f=fact_rec(n)
    print("Factorial of {} is: {}".format(n,f))
```

Enter the number:5
Factorial of 5 is: 120

```
In [13]: #To generate N terms of Fibonacci series(Iterative method)
def Fibonacci_iter():
    nterms = int(input("Enter the number of terms:"))
    #First two terms
    n1 = 0
    n2 = 1
    count = 0
    #check if the number of terms is valid
    if nterms <=0:
        print("Please enter a positive number")
    elif nterms == 1:
        print("Fibonacci sequence upto",nterms,":")
        print(n1)
    else:
        print("Fibonacci sequence upto",nterms,":")
    while count < nterms:
        print(n1,end=' , ')
        nth = n1 + n2
        #update values
        n1 = n2
        n2 = nth
        count += 1

    print(Fibonacci_iter())
```

Enter the number of terms:10
Fibonacci sequence upto 10 :
0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , None

```
In [17]: #To generate N terms of Fibonacci series(recursive method)
def Fibonacci_rec(q):
    if q == 1:
        return 0
    elif q == 2:
        return 1
    else:
        return Fibonacci_rec(q-1) + Fibonacci_rec(q-2)

m = int(input("Enter the number of terms: "))
for Num in range(1, m+1):
    print(Fibonacci_rec(Num),end=' ')
```

Enter the number of terms: 10
0 1 1 2 3 5 8 13 21 34

```
In [ ]:
```

```
In [ ]:
```

Name:Syed Mohammad Wazidul Ifkat

USN:19BTRCY069

LAB PROGRAM 3

3(A)Write python programs to print all prime numbers between N and M

In [2]:

```
#Method1
num = int(input("Enter a number: "))
if num > 1:
    for i in range(2,num):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",num//i,"is",num)
            break
        else:
            print(num,"is a prime number")
    else:
        print(num,"is not a prime number")
```

Enter a number: 97
97 is a prime number

In [3]:

```
#Method2
num = int(input("Enter a number: "))
if num > 1:
    for i in range(2,(num//2)+1):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",num//i,"is",num)
            break
        else:
            print(num,"is a prime number")
    else:
        print(num,"is not a prime number")
```

Enter a number: 99
99 is not a prime number
3 times 33 is 99

In [5]:

```
#Method3
import math

num = int(input("Enter a number: "))

if num > 1:
    for i in range(2,math.floor(math.sqrt(num))):
        if (num % i) == 0:
            print(num,"is not a prime number")
            print(i,"times",num//i,"is",num)
            break
        else:
            print(num,"is a prime number")
    else:
        print(num,"is not a prime number")
```

Enter a number: 97
97 is a prime number

In [25]:

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter upper range: "))

print("Prime numbers between",lower,"and",upper,"are: ")
for num in range(lower,upper + 1):
    if num > 1:
        for i in range (2, num):
            if((num % i) == 0):
                break
        else:
            print(num,end=',')
```

Enter lower range: 2
Enter upper range: 10
Prime numbers between 2 and 10 are:
2,3,5,7,

In [26]:

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter upper range: "))

print("Prime numbers between",lower,"and",upper,"are: ")
for num in range(lower,upper + 1):
    if num > 1:
        for i in range (2, (num//2)+1):
            if((num % i) == 0):
                break
        else:
            print(num,end=',')
```

Enter lower range: 10
Enter upper range: 20
Prime numbers between 10 and 20 are:
11,13,17,19,

In [24]:

```
#prime numbers upto n
n = int(input("Enter limit: "))

print("Prime numbers upto",n,"are: ")
for num in range (2, n + 1):
    if num > 1:
        for i in range(2, (num//2)+1):
            if num % i == 0:
                break
        else:
            print(num,end=',')
```

Enter limit: 10
Prime numbers upto 10 are:
5,7,7,9,

3(B)Write python program to find GCD for two numbers,input by user

In [14]:

```
def gcd(a,b):
    while(a!=b):
        if(a>b):
            a=a-b
        else:
            b=b-a
    return a

a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
g=gcd(a,b)
print("GCD of {} and {} is:{}".format(a,b,g))
```

Enter first number:10
Enter second number:25
GCD of 10 and 25 is:5

In [23]:

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)

a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
print("GCD of {} and {} is:{}".format(a,b,GCD))
```

Enter first number:10
Enter second number:25
GCD of 10 and 25 is:5

3(C)Write a python program to find largest among three numbers,input by user

In [17]:

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))
large=a
if(b>large):
    large=b
if(c>large):
    large=c
print("Largest of {},{} and {} is:{}".format(a,b,c,large))
```

Enter the first number:1
Enter the second number:2
Enter the third number:3
Largest of 1,2 and 3 is:3

In [18]:

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))

if (a > b) and (a > c):
    large = a
elif (b > a) and (b > c):
    large = b
print("Largest of {},{} and {} is:{}".format(a,b,c,large))
```

Enter the first number:1
Enter the second number:2
Enter the third number:3
Largest of 1,2 and 3 is:3

In []:

Name: Syed Mohammad Wazidul Ifkat

USN: 19BTRCY069

LAB PROGRAM: 4

4(A) Elaborate string Operations such as string declaration, traversing, slicing, concatenating and sorting

```
In [3]: #String Declaration

#creating a string with single quotes
String1 = "Jain Deemed-to-be University 19BTRCY069"
print("string with the use of single Quotes: ")
print(String1)

#creating a string with double quotes
String1 = "Jain Deemed-to-be University 19BTRCY069"
print("\nString with the use of double Quotes: ")
print(String1)

#creating a string with triple quotes
String1 = "Jain Deemed-to-be University 19BTRCY069"
print("\nString with the use of triple Quotes: ")
print(String1)

# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
               the string in Python
               Fifth Sem IoT Students"""
print(my_string)
```

string with the use of single Quotes:
Jain Deemed-to-be University 19BTRCY069

String with the use of double Quotes:
Jain Deemed-to-be University 19BTRCY069

String with the use of triple Quotes:
Jain Deemed-to-be University 19BTRCY069
Hello, welcome to
 the string in Python
 Fifth Sem IoT Students

```
In [5]: #string Traversing

string_name = "19BTRCY069"

#iterate over index
for element in range(0, len(string_name)):
    print(string_name[element])
```

1
9
B
T
R
C
Y
0
6
9

```
In [6]: #string slicing
String = 'ASTRING'

#using slice constructor
s1 = slice(3)
```

```
s2 = slice(1,5,2)
s3 = slice(-1, -12, -2)

print("string slicing")
print(String[s1])
print(String[s2])
print(String[s3])
```

string slicing
AST
SR
GITA

In [7]:

```
#STRING CONCETANING

var1 = "Hello "
var2 = "World"

# + operator is used to combine strings
var3 = var1 + var2
print(var3)

#string concatenation
var1 = "Hello"
var2 = "World"

# % operator is used here to combine the string
print("\n\n% s % s" % (var1,var2))
```

Hello World

Hello World

In [8]:

```
#STRING SORTING

#using join() + sorted()

#initializing string
test_string = "Syed Ifkat"

#printing original string
print("The original String : " + str(test_string))

#sorting a string
res = ''.join(sorted(test_string))

print("String after sorting : " + str(res))
```

The original String : Syed Ifkat
String after sorting : ISadefkty

B.Implement a python script to check the element is in the list or not by using linear search and Binary search.

Linear Search

In [12]:

```
def search(list,n):

    for i in range(len(list)):
        if list[i] == n:
            return True
    return False

#list which contains both string and numbers
list = ['python','69', 'Linear search', 4,'Lab-4', 'CSE-IOT']

#Driver Code
n = 'Lab-4'

if search(list, n):
    print("Found")
```

```
else:
    print("Not Found")
```

Found

Binary Search

```
In [14]: #Iterative Binary Search Function
def binary_search(arr,x):
    low = 0
    high = len(arr) - 1
    mid = 0

    while low <= high:

        mid = (high + low) // 2

        #If x is greater,ignore left half
        if arr[mid] < x:
            low = mid + 1

        #if x is smaller,ignore right half
        elif arr[mid] > x:
            high = mid - 1

        #means x is present at mid
        else:
            return mid
    #if we reach here,then the element was not present
    return -1

#TEXT ARRAY
arr = [2,3,4,10,40]
x = 10

#function call
result = binary_search(arr,x)

if result != -1:
    print("Element is present at index",str(result))
else:
    print("Element is not present in array")
```

Element is not present in array

C.Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques

```
In [18]: #BUBBLE SORT

def bubbleSort(arr):
    n = len(arr)

    #traverse through all array elements
    for i in range(n-1):

        #last i elements are already in place
        for j in range (0, n-i-1):

            #traverse the array from 0 to n-i-1
            #swap the element found is greater than the next element
            if arr[j] > arr[j + 1]:
                arr[j],arr[j + 1] = arr[j + 1],arr[j]

# Driver code to test above
arr = [64,34,25,12,22,11,90]

bubbleSort(arr)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" % arr[i]),
```

Sorted array is:

11
12

22
25
34
64
90

```
In [19]: #Selection Sort
import sys
A = [64,25,12,22,11]

#traverse through all array elements
for i in range(len(A)):

    #Find the minimum element in remaining
    #unsorted array
    min_idx = i
    for j in range(i+1, len(A)):
        if A[min_idx] > A[j]:
            min_idx = j

    A[i],A[min_idx] = A[min_idx],A[i]

#Driver code to test above
print("sorted array")
for i in range(len(A)):
    print("%d" %A[i]),
```

sorted array
11
22
12
25
64

```
In [20]: #insertion sort

def insertionSort(arr):

    #traverse through 1 to len(arr)
    for i in range(1,len(arr)):

        key = arr[i]

        #Move elements of arr[0..i-1]

        j = i-1
        while j >= 0 and key < arr[j] :
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

#Driver code to test above
arr = [12,11,13,5,6]
insertionSort(arr)
for i in range(len(arr)):
    print ("% d" % arr[i])
```

5
6
11
12
13

```
In [29]: #MERGE SORT
def mergeSort(arr):
    if len(arr) > 1:

        #finding the mid of the array
        mid = len(arr)//2

        #Dividing the array elements
        L = arr[:mid]

        #into 2 halves
        R = arr[mid:]

        #Sorting the first half
        mergeSort(L)
```

```

#Sorting the second half
mergeSort(R)

i = j = k = 0

#copy data to temp arrays L[] and r[]
while i < len(L) and j < len(R):
    if L[i] < R[j]:
        arr[k] = L[i]
        i += 1
    else:
        arr[k] = R[j]
        j += 1
    k += 1

#checking if any element was left
while i < len(L):
    arr[k] = L[i]
    i += 1
    k += 1

while j < len(R):
    arr[k] = R[j]
    j += 1
    k += 1

# code to print the list

def printlist(arr):
    for i in range(len(arr)):
        print(arr[i],end=" ")
    print()

#driver code
if __name__ == '__main__':
    arr = [12,11,13,5,6,7]
    print("Given array is",end="\n")
    printlist(arr)
    mergeSort(arr)
    print("Sorted array is: ",end="\n")
    printlist(arr)

```

Given array is
12 11 13 5 6 7

```

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-29-cd83658f570f> in <module>
     53     print("Given array is",end="\n")
     54     printlist(arr)
--> 55     mergeSort(arr)
     56     print("Sorted array is: ",end="\n")
     57     printlist(arr)

<ipython-input-29-cd83658f570f> in mergeSort(arr)
     13
     14     #Sorting the first half
--> 15     mergeSort(L)
     16
     17     #Sorting the second half

<ipython-input-29-cd83658f570f> in mergeSort(arr)
     16
     17     #Sorting the second half
--> 18     mergeSort(R)
     19
     20     i = j = k = 0

<ipython-input-29-cd83658f570f> in mergeSort(arr)
     22     #copy data to temp arrays L[] and r[]
     23     while i < len(L) and j < len(R):
--> 24         if L[i] < R[j]:
     25             arr[k] = L[i]
     26             i += 1

TypeError: 'list' object is not callable

```

In []:

Name: Syed Mohammad Wazidul Ifkat

USN: 19BTRCY069

LAB PROGRAM 5

5(A) Implement python script to show the usage of various available in python language.

```
In [3]: print("arithmetic Operators: ")

x = 20
y = 10

print('x + y =', x+y)
print('x - y =', x-y)
print('x * y =', x*y)
print('x / y =', x/y)
print('x // y =', x//y)
print('x ** y =', x**y)

print("Comparison Operators: ")
x = 5
y = 10

print('x > y =', x>y)
print('x < y =', x<y)
print('x == y =', x==y)
print('x != y =', x!=y)
print('x >= y =', x>=y)
print('x <= y =', x<=y)

print("Logical Operators: ")
x = True
y = False

print('x and y is', x and y)
print('x or y is', x or y)
print('not x is', not x)

print("Identity Operators: ")
x = ["orange", "apple"]
y = ["orange", "apple"]
z = x

print(x is z)
print(x is not z)

print("Membership Operators: ")
x = ["orange", "banana"]

print("banana" in x)
print("apple" not in x)
```

```
arithmetic Operators:
x + y = 30
x - y = 10
x * y = 200
x / y = 2.0
x // y = 2
x ** y = 10240000000000
```

Comparison Operators:

x > y = False

x < y = True

x == y = False

x != y = True

x >= y = False

x <= y = True

Logical Operators:

x and y is False

x or y is True

not x is False

Identity Operators:

True

False

Membership Operators:

True

True

5(B) Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs.12.00 per hour for every hour worked above 40 hours. Assume that employee do not work for fractional part of an hour.

In [7]:

```
overtime_pay = overtime_pay_total = 0
for i in range(1,11):
    time_worked = int(input(f"Enter the time employee {i} worked in hr: "))
    if time_worked > 40:
        over_time = time_worked - 40
        overtime_pay = (12 * over_time)
        print(f"Overtime Pay of employees {i} is", overtime_pay)
        overtime_pay_total += overtime_pay
    else:
        print(f"Overtime Pay of employees {i} is 0")

print(f"\nTotal Overtime pay of 10 employees is", overtime_pay_total)
```

```
Enter the time employee 1 worked in hr: 30
Overtime Pay of employees 1 is 0
Enter the time employee 2 worked in hr: 40
Overtime Pay of employees 2 is 0
Enter the time employee 3 worked in hr: 50
Overtime Pay of employees 3 is 120
Enter the time employee 4 worked in hr: 60
Overtime Pay of employees 4 is 240
Enter the time employee 5 worked in hr: 70
Overtime Pay of employees 5 is 360
Enter the time employee 6 worked in hr: 80
Overtime Pay of employees 6 is 480
Enter the time employee 7 worked in hr: 90
Overtime Pay of employees 7 is 600
Enter the time employee 8 worked in hr: 85
Overtime Pay of employees 8 is 540
Enter the time employee 9 worked in hr: 75
Overtime Pay of employees 9 is 420
Enter the time employee 10 worked in hr: 65
Overtime Pay of employees 10 is 300

Total Overtime pay of 10 employees is 3060
```

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Name: Syed Mohammad Wazidul Ifkat

USN: 19BTRCY069

LAB PROGRAM-6

6(A) Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

```
In [1]: import pandas as pd
import numpy as np

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin'],
             'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
             'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
             'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index=labels)
print(df)
```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

6(B) To create a 4X2 integer array and prints its attributes

```
In [2]: import numpy

sampleArray = numpy.array([[3, 6, 9, 12], [15, 18, 21, 24],
                           [27, 30, 33, 36], [39, 42, 45, 48], [51, 54, 57, 60]])
print("Printing Input Array")
print(sampleArray)

print("\n Printing array of odd rows and even columns")
newArray = sampleArray[::2, 1::2]
print(newArray)
```

Printing Input Array

```
[[ 3  6  9 12]
 [15 18 21 24]
 [27 30 33 36]
 [39 42 45 48]
 [51 54 57 60]]
```

Printing array of odd rows and even columns

```
[[ 6 12]
 [30 36]
 [54 60]]
```

6(C) For the given numpy array return array of odd rows and even columns

```
In [3]: import numpy as np
```

```
arr = np.array([[1, 2, 3], [4, 5, 6],[7, 8, 9],[10, 11, 12],[13, 14, 15]])

case1=arr[::2,:]    #odd rows
case2=arr[1::2,:]    #even rows
case3=arr[:,::2]    #odd cols
case4=arr[:,1::2]    #even cols
print(case1)
print("\n")
print(case2)
print("\n")
print(case3)
print("\n")
print(case4)
print("\n")
```

```
[[ 1  2  3]
 [ 7  8  9]
 [13 14 15]]
```

```
[[ 4  5  6]
 [10 11 12]]
```

```
[[ 1  3]
 [ 4  6]
 [ 7  9]
 [10 12]
 [13 15]]
```

```
[[ 2]
 [ 5]
 [ 8]
 [11]
 [14]]
```

6(D)To add the two NumPy arrays and modify a result array by calculating the square root of each element.

In [4]:

```
import numpy

arrayOne = numpy.array([[5, 6, 9], [21,18, 27]])
arrayTwo = numpy.array([[15,33, 24], [4,7, 1]])

resultArray = arrayOne + arrayTwo
print("addition of two arrays is \n")
print(resultArray)

for num in numpy.nditer(resultArray, op_flags = ['readwrite']):
    num[...] = num*num
print("\nResult array after calculating the square root of all elements\n")
print(resultArray)
```

addition of two arrays is

```
[[20 39 33]
 [25 25 28]]
```

Result array after calculating the square root of all elements

```
[[ 400 1521 1089]
 [ 625  625  784]]
```

In []:

Name: Syed Mohammad Wazidul Ifkat

USN: 19BTRCY069

LAB PROGRAM-7

7(A) To read total profit of all months and show it using a line plot where x axis should be month number and y axis should be named as total profit

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("Company_sales_data.csv")

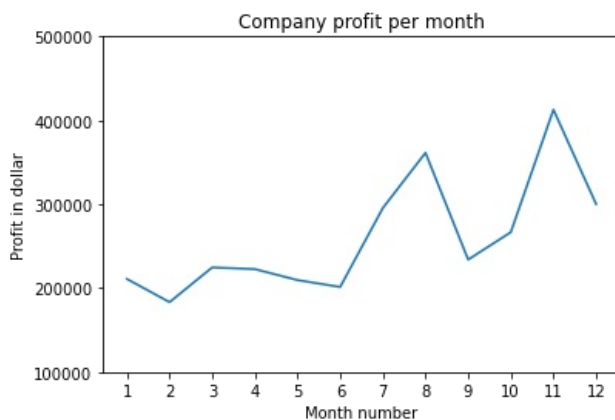
df.head()
```

```
Out[3]:
```

	month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600

```
In [4]: import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("Company_sales_data.csv")
profitList = df['total_profit'].tolist()
monthList = df['month_number'].tolist()
plt.plot(monthList, profitList, label = 'Month-wise Profit data of last year')
plt.xlabel('Month number')
plt.ylabel('Profit in dollar')
plt.xticks(monthList)
plt.title('Company profit per month')
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



7(B) To get total profit of all months and show line plot with the following style properties:

Line Style dotted and Line-color should be red

Show legend at the lower right location.

X label name = Month Number

Y label name = Sold units number

Add a circle marker.

In [5]:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("Company_sales_data.csv")
profitList = df ['total_profit'].tolist()
monthList = df ['month_number'].tolist()

plt.plot(monthList, profitList, label = 'Profit data of last year',
         color='r', marker='o', markerfacecolor='k',
         linestyle='--', linewidth=3)

plt.xlabel('Month Number')
plt.ylabel('Profit in dollar')
plt.legend(loc='lower right')
plt.title('Company Sales data of last year')
plt.xticks(monthList)
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

NAME:SYED MOHAMMAD WAZIDUL IFKAT

USN:19BTRCY069

LAB PROGRAM-8

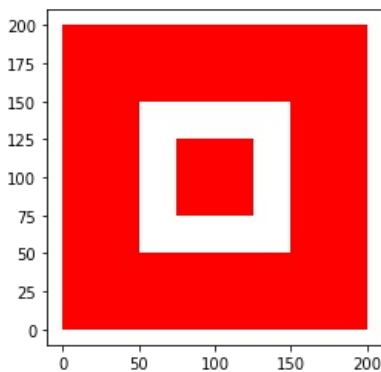
A.Draw the target symbol(a set of concentric squares,alternating red and white)in a graphics window,that is,200 pixels wide by 200 pixels high.

```
In [1]: import matplotlib.pyplot as plt
rect=plt.Rectangle((0,0),200,200,fc='r')
plt.gca().add_patch(rect)

rect=plt.Rectangle((50,50),100,100,fc='w')
plt.gca().add_patch(rect)

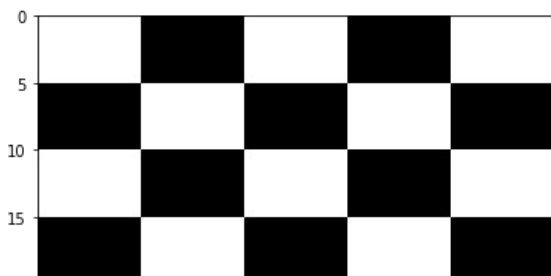
rect=plt.Rectangle((75,75),50,50,fc='r')
plt.gca().add_patch(rect)

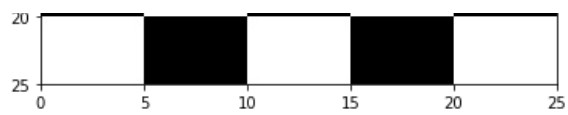
plt.axis('scaled')
plt.show()
#plt.figure()
```



B.Create a 5x5 rectangle whose top left corner is at (row5, col5).If the sum of the rows and columns number is even,set the fill color of the rectangle to white,otherwise set it to the black.Then draw the rectangle.

```
In [2]: plt.figure()
plt.axis([0,25,25,0])
curr=plt.gca()
for i in range(0,5):
    for j in range(0,5):
        if((i+j)%2==0):
            curr.add_patch(plt.Rectangle((i*5,j*5),5,5,color='white'))
        else:
            curr.add_patch(plt.Rectangle((i*5,j*5),5,5,color='black'))
plt.show()
```





In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Name: Syed Mohammad Wazidul Ifkat

USN: 19BTRCY069

Subject: Python Workshop Lab

LAB PROGRAM : 9

1. Introduction to Raspberry Pi board.

Raspberry Pi, developed by Raspberry Pi Foundation in association with Broadcom, is a series of small single-board computers and perhaps the most inspiring computer available today.

From the moment you see the shiny green circuit board of Raspberry Pi, it invites you to tinker with it, play with it, start programming, and create your own software with it. Earlier, the Raspberry Pi was used to teach basic computer science in schools but later, because of its low cost and open design, the model became far more popular than anticipated.

It is widely used to make gaming devices, fitness gadgets, weather stations, and much more. But apart from that, it is used by thousands of people of all ages who want to take their first step in computer science.

It is one of the best-selling British computers and most of the boards are made in the Sony factory in Pencoed, Wales.

Generations and Models

In 2012, the company launched the Raspberry Pi and the current generations of regular Raspberry Pi boards are **Zero, 1, 2, 3, and 4**.

Generation 1 Raspberry Pi had the following four options –

- Model A
- Model A +
- Model B
- Model B +

Among these models, the **Raspberry Pi B models** are the original credit-card sized format.

On the other hand, the **Raspberry Pi A models** have a smaller and more compact footprint and hence, these models have the reduced connectivity options.

Raspberry Pi Zero models, which come with or without GPIO (general-purpose input output) headers installed, are the most compact of all the Raspberry Pi boards types.

2. Setting up Python on Raspberry Pi board.

he steps to install Python on Raspberry Pi 4 are as follows-

1. Update the Raspbian

Update the Raspbian before installing python.

```
sudo apt-get update
```

2. Prerequisites

Before installing Python 3.8 there are some dependencies that we need to install. Use the following command to install the required dependencies.

```
sudo apt-get install -y build-essential tk-dev libncurses5-dev libncursesw5-dev  
libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev  
libexpat1-dev liblzma-dev zlib1g-dev libffi-dev tar wget vim
```

3. Download Python

You can download Python from the official [website](https://www.python.org/) or use the following command.

```
wget https://www.python.org/ftp/python/3.8.0/Python-3.8.0.tgz
```

4. Install Python 3.8

Now we will extract and install Python from the source.

```
sudo tar xzf Python-3.8.0.tgz
```

```
cd Python-3.8.0
```

```
sudo ./configure --enable-optimizations
```

```
sudo make -j 4
```

```
sudo make altinstall
```

<https://installvirtual.com/how-to-connect-raspberry-pi-to-wifi-without-a-monitor/>

5. Check Python version

Now Python is installed you can check the version using the following command.

```
python3.8 -V
```

6. Make Python 3.8 as the default version

If you want to use python 3.8 as a default version you can create an alias. `echo "alias python=/usr/local/bin/python3.8" >> ~/.bashrc`

Then source the .bashrc file.

```
source ~/.bashrc
```

7. Check Python Version

After creating an alias check the python version again.

```
python -V  
Python 3.8.0
```

Now you have successfully installed Python 3.8 on Raspberry Pi.

8. Clean up

Now you can clean up using the archive

```
sudo rm -rf Python-3.8.0.tgz  
sudo rm -rf Python-3.8.0
```

c. Running Python from a file.

Running a Python Program from a File

You can individually enter and run commands one line at a time in the Python interpreter, which is incredibly useful for trying out different commands (or using it as a calculator!). Often, you will want to save your commands together in one or more files so that you can run them all at once.

The simplest way to do this is to create a file from the terminal, although you are welcome to use the Raspbian graphical editor, Leafpad, as well (found under Accessories > Text Editor when you click on the Raspberry Pi icon in the top left).

Still in a terminal, enter the command:

```
nano hello.py
```

D.Printing Hello world from file.

This creates a file named hello.py in your home directory (/home/pi) and starts editing it with the nano program.

In this file, enter the following on the first line:

```
print("Hello, World!")
```