

DELHI PUBLIC SCHOOL
KAMPTEE ROAD, NAGPUR



SUBMITTED BY	MARUVAPALLI MANOJ
CLASS	12TH 'A'
TOPIC	PRACTICAL FILE
SUBMITTED TO	MR CHANDRASHAKER UPRADE

CERTIFICATE

This is to certify that Maruvapalli Manoj, a student of class 12th 'A' has successfully completed the project on the topic of Python Interface with SQL databases under the guidance of Mr Chandrashaker Upgrade during session 2019-20 at Delhi Public school, Kamptee Road, Nagpur

Signature of External
Examiner

Signature of Computer
Science Teacher

ACKNOWLEDGEMENT

I place my sincere thanks to my computer science teacher Mr Chandrashaker Uprade for his guidance and advices to complete my work succesfully.

I also thank our principal Mrs Ritu Sharma for providing me all the facilities to finish the project on time.

Last but not least I thank my parents for their encouragement and support in my humble venture.

Index

1.PYTHON	-----	6
2.MYSQL	-----	42

PYTHON

Program to check weather a number is prime or not.

```
x=int(input("Enter a number:"))
y=int(x/2)+1
for i in range (2,y):
    rem=x%i
    if rem==0:
        print("Number",x,"is not a prime number.")
        break
else:
    print("Number",x,"is a prime number.")
```

OUTPUT:

```
Enter a number:59
Number 59 is a prime number.
```

Program to check a number is palindrome or not.

```
def cekpal(n):  
    num=n  
    d=0  
    rev=0  
    while n>0:  
        d=n%10  
        n=int(n/ 10)  
        rev=int(rev*10+d)  
    if(num==rev):  
        return True  
    else:  
        return False  
  
#__main__  
n=int(input("Enter a number:"))  
if cekpal(n):  
    print(n,"is a palindrome")  
else:  
    print(n,"is a not palindrome")
```

OUTPUT:

```
Enter a number:123454321
123454321 is a palindrome
```


#Program to calculate compound interest.

```
def compint(p,r,n):  
    m=n  
    x=p*(1+r/100)**(n)  
    return x  
  
p=float(input("Enter the principal amount:"))  
r=float(input("Enter the rate of intrest:"))  
n=float(input("Enter the number of years:"))  
x=compint(p,r,n)  
print("The amount after",n,"years is",x,"rupees.")
```

OUTPUT:

```
Enter the principal amount:6250  
Enter the rate of intrest:8  
Enter the number of years:2  
The amount after 2.0 years is 7290.000000000001 rupees.
```

Program to display ASCII code of a character and vice versa.

```
def asc(a):  
    print([ord(c) for c in a])  
    def asc_val(a):  
        char=[chr(ascii) for ascii in a]  
        ".join(char)  
        print(char)  
#__main__  
print("Enter '1' for converting string in ascii code")  
print("Enter '2' for converting ascii code in string")  
x=int(input("Enter your choice:"))  
if x==1:  
    n=input("Enter your character:")  
    asc(n)  
elif x==2:  
    n=eval(input("Enter your ASCII value:"))  
    asc_val(n)  
else:  
    print("You have entered a in valid choice")
```

OUTPUT:

```
Enter '1' for converting string in ascii code
Enter '2' for converting ascii code in string
Enter your choice:2
Enter your ASCII value:[11,22,33,44,55,66,77,8,8,8,9,9,0,128]
['\x0b', '\x16', '!', ',', '7', 'B', 'M', '\x08', '\x08', '\x08', '\t', '\t', '\x00', '\x80']
```

Program to input a character and print weather a given character is an alphabet, digit or any other character.

```
ch = input("Please Enter Your Own Character : ")
if((ch >= 'a' and ch <= 'z') or (ch >= 'A' and ch <= 'Z')):
    print("The Given Character ", ch, "is an Alphabet")
elif(ch >= '0' and ch <= '9'):
    print("The Given Character ", ch, "is a Digit")
else:
    print("The Given Character ", ch, "is Not an Alphabet or a Digit")
```

OUTPUT:

```
Please Enter Your Own Character : 55
The Given Character  55 is a Digit
```

Program to calculate the factorial of an integer using recursion.

```
def factorial(n):  
    if n == 1:  
        return n  
    else:  
        return n*factorial(n-1)  
num = int(input("Enter a number: "))  
if num < 0:  
    print("Sorry, factorial does not exist for negative numbers")  
elif num == 0:  
    print("The factorial of 0 is 1")  
else:  
    print("The factorial of",num,"is",factorial(num))
```

OUTPUT:

```
Enter a number: 5  
The factorial of 5 is 120
```

Program to print Fibonacci series using recursion.

```
def fib(n):  
    if n==1:  
        return 0  
    elif n==2:  
        return 1  
    else:  
        return fib(n-1)+fib(n-2)  
  
#__main__  
n=int(input("Enter the last term required:"))  
for i in range(1,n+1):  
    print(fib(i),end=',')  
print ("...")
```

OUTPUT:

```
Enter the last term required:8  
0,1,1,2,3,5,8,13,...
```

Program for binary search.

```
def binarysearch(ar,key):  
    low=0  
    high=len(ar)-1  
    while low<=high:  
        mid=int((low+high)/2)  
        if key==ar[mid]:  
            return mid  
        elif key<ar[mid]:  
            high=mid-1  
        else:  
            low=mid+1  
    else:  
        return-999  
  
#__main__  
ar=eval(input("Enter a list of integers:" ))  
item=int(input("Enter search item:"))  
res=binarysearch(ar,item)  
if res>=0:  
    print(item,"found at index",res,".")  
else:  
    print("Sorry",item,"not found in array.")
```

OUTPUT:

```
Enter a list of integers:[12,15,21,25,28,32,33,36,43,45]  
Enter search item:32  
32 found at index 5 .
```


#Program to find whether a string is
palindrome or not using recursion.

```
def par(s):  
    if len(s) < 1:  
        return True  
    else:  
        if s[0] == s[-1]:  
            return par(s[1:-1])  
        else:  
            return False  
a=str(input("Enter string:"))  
if (par(a)==True):  
    print("String is a palindrome")  
else:  
    print("String isn't a palindrome")
```

OUTPUT:

```
Enter string:mam  
String is a palindrome
```

#Program to count the number of vowels present in a text file.

```
file1=open("D:\\mytext.txt","r")
str1=file1.read()
vowel_count=0
for i in str1:
    if(i=='A' or i=='a' or i=='E' or i=='e' or i=='I' or i=='i' or i=='O' or
i=='o' or i=='U' or i=='u'):
        vowel_count+=1
print("The number of vowels in the text file :",vowel_count)
file1.close()
```

OUTPUT:

```
The number of vowels in the text file : 1635
```

#Program to write those lines which have the character 'p' from one text file to another text file.

```
fn=open("D:\\mytext.txt","r")
fn1=open("D:\\mytext1.txt","w")
count=fn.read()
type(count)
for i in range (0,len(count)):
    if count[i]=='p':
        fn1.write(count[i])
    else:
        pass
fn1.close
fn1=open("D:\\mytext1.txt","r")
count1=fn1.read()
print(count1)
fn.close
fn1.close
```

OUTPUT:

```


ppppppppppppppppppppppppppppppppppppp  

    <function TextIOWrapper.close()>


```

#Program to count number of words in a file.

```
num_words=0
with open("D:\\mytext.txt","r") as f:
    for line in f:
        words=line.split()
        num_words+=len(words)
print("Number of words:",num_words)
```

OUTPUT:

Number of words: 11

#Program to calculate the value of sin(x) using its Taylor series expansion upto n terms.

```
import math
def cal_sin(n):
    accuracy=0.0001
    n=n*(3.142/180.0)
    x=n
    sinx=n
    sinval=math.sin(n)
    i=1
    while (True):
        denominator=2*i*(2*i+1)
        x=-x*n*n/denominator
        sinx=sinx+x
        i=i+1
        if(accuracy<= abs(sinval-sinx)):
            break
    print(sinx)
n=int(input("Enter the angle:"))
cal_sin(n)
```

OUTPUT:

```
Enter the angle:45  
0.7047230747708333
```

#Program to generate random numbers between 1 to 6 and check whether the user has won a lottery or not.

```
import random
x=int(input("Enter a number :"))
a=int(random.randint(0,6))
print("The lottery number is",a)
if(x==a):
    print ("You WON!!!!")
else:
    print("YOU LOSE")
```

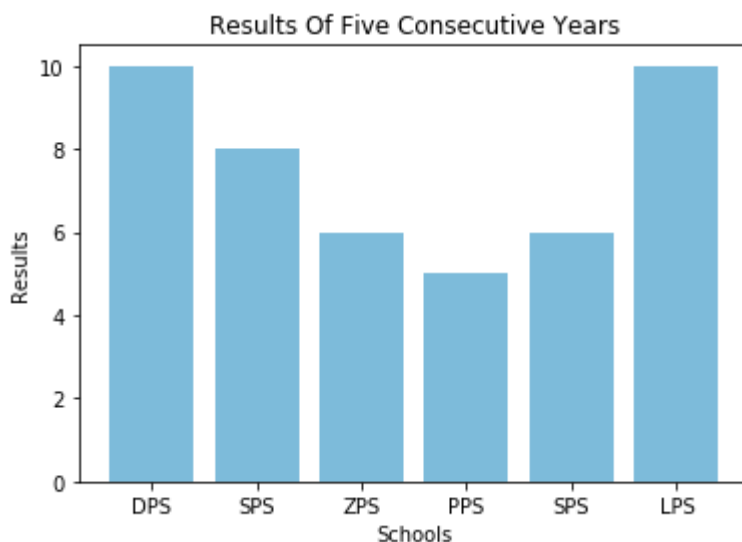
OUTPUT:

```
Enter a number :5
The lottery number is 5
You WON!!!!
```

#Program to plot a bar chart on to display the result of a school for five consecutive years.

```
import matplotlib.pyplot as plt
import numpy as np
objects = ('DPS', 'SPS', 'ZPS', 'PPS', 'SPS', 'LPS')
y = np.arange(len(objects))
performance = [10,8,6,5,6,10]
plt.bar(y, performance, align='center', alpha=0.5)
plt.xticks(y, objects)
plt.ylabel('Results')
plt.xlabel('Schools')
plt.title('Results Of Five Consecutive Years')
plt.show()
```

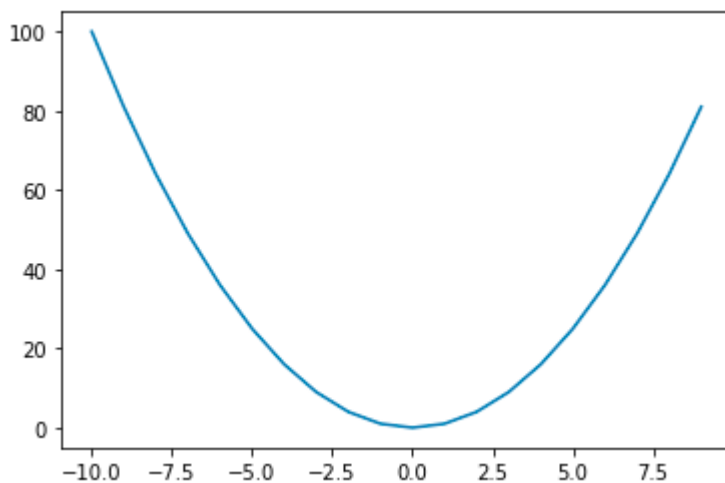
OUTPUT:



#Program to plot a graph for the function
 $y=x^2$.

```
import matplotlib.pyplot as plt  
import numpy as np  
x = np.arange( -10 , 10 )  
y = np.square( x )  
plt.plot( x , y )  
plt.show()
```

OUTPUT:



#Program for linear search.

```
def linear_search(alist,key):  
    for i in range (len(alist)):  
        if alist[i]==key:  
            return True  
    return False  
  
alist=eval(input("enter the list of numbers:"))  
key=str(input("The number to search for:"))  
index=linear_search(alist,key)  
if index<0:  
    print(key,"was not found in the list")  
else:  
    print(key,"was found in the list")
```

OUTPUT:

```
enter the list of numbers:[15,6,13,22,3,52,2]  
The number to search for:22  
22 was found in the list
```

#Program for bubble sort search.

```
aList=eval(input("Enter Your List:"))
print("Orginal list is:",aList)
n=len(aList)
for i in range(n):
    for j in range (0,n-i-1):
        if aList[j]>aList[j+1]:
            aList[j],aList[j+1]=aList[j+1],aList[j]
print("List after sorting:",aList)
```

OUTPUT:

```
Enter Your List:[15,6,13,22,3,52,2]
Orginal list is: [15, 6, 13, 22, 3, 52, 2]
List after sorting: [2, 3, 6, 13, 15, 22, 52]
```

#Menu based program to perform the operation on stack in python.

```
def empty (stk):  
    if stk==[]:  
        return True  
    else:  
        return False  
  
def push (stk,item):  
    stk.append(item)  
    top=len(stk)-1  
  
def pop(stk):  
    if empty(stk) is True:  
        print("empty stack")  
    else:  
        item=stk.pop()  
        if len(stk)==0:  
            top=None  
        else:  
            top=len(stk)-1  
            print(top,item)  
        return item  
  
def peek(stk):  
    if empty:
```

```

        return "underflow"
    else:
        top=len(stk-1)
        return stk[top]
def display(stk):
    if empty(stk):
        print("empty stack")
        quit
    else :
        top=len(stk)-1
        print(stk[top],"top")
        for i in range(top,-1,-1):
            print(stk[i])
#main program
stk=[1,2,3,4,5,6,7,8,9]
top = None
while empty(stk) is not True:
    print("STACK OPERATION")
    print("1.PUSH")
    print("2.POP")
    print("3.PEEK")
    print("4.DISPLAY")
    print("5.EXIT")
    ch=int(input("ENTER YOUR CHOICE FROM 1 TO 5: "))
    if ch not in (1,2,3,4,5):

```

```
print ("invalid choice")

    break
elif ch == 1:

    item=int(input("enter item:"))
    push(stk,item)
elif ch==2:
    item= pop(stk)
    if item=="underflow":
        print("underflow the stack is empty")
        print(stk)
    else:
        print("poped item is ",item)
elif ch==3:
    item=peek(stk)
    if item=="underflow":
        print("underflow,stack is empty")
    else:
        print ("top most item is",item)
elif ch==4:
    display(stk)
elif ch==5:
    break
```

OUTPUT:

```
STACK OPERATION
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
ENTER YOUR CHOICE FROM 1 TO 5: 1
enter item:0
STACK OPERATION
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
ENTER YOUR CHOICE FROM 1 TO 5: 2
8 0
poped item is 0
STACK OPERATION
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
ENTER YOUR CHOICE FROM 1 TO 5: 3
underflow,stack is empty
STACK OPERATION
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
ENTER YOUR CHOICE FROM 1 TO 5: 4
9 ,top
9
8
7
6
5
4
3
2
1
STACK OPERATION
1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
ENTER YOUR CHOICE FROM 1 TO 5: 5
```

Menu based program to perform the operation on queue in python.

```
class Queue:

    def __init__(self):

        self.items = []

    def is_empty(self):

        return self.items == []

    def enqueue(self, data):

        self.items.append(data)

    def dequeue(self):

        return self.items.pop(0)

q = Queue()
while True:

    print('enqueue <value>')

    print('dequeue')

    print('quit')

    do = input('What would you like to do? ').split()
```



```
operation = do[0].strip().lower()
if operation == 'enqueue':
    q.enqueue(int(do[1]))
elif operation == 'dequeue':
    if q.is_empty():
        print('Queue is empty.')
    else:
        print('dequeued value: ',q.dequeue())
elif operation=='quit':
    break
```

OUTPUT:

```
enqueue <value>
dequeue
quit
What would you like to do? 3
enqueue <value>
dequeue
quit
What would you like to do? quit
```

Menu based program for circular queue in python.

```
class CircularQueue():  
    def __init__(self, size): # initializing the class  
        self.size = size  
  
        self.queue = [None for i in range(size)]  
  
        self.front = self.rear = -1  
  
    def enqueue(self, data):  
        if ((self.rear + 1) % self.size == self.front):  
            print(" Queue is Full\n")  
        elif (self.front == -1):  
            self.front = 0  
            self.rear = 0  
            self.queue[self.rear] = data  
        else:  
            self.rear = (self.rear + 1) % self.size  
            self.queue[self.rear] = data  
  
    def dequeue(self):  
        if (self.front == -1):  
            print ("Queue is Empty\n")  
        elif (self.front == self.rear):  
            temp=self.queue[self.front]  
            self.front = -1
```

```

        self.rear = -1

    return temp

else:

    temp = self.queue[self.front]

    self.front = (self.front + 1) % self.size

    return temp

def display(self):

    if(self.front == -1):

        print ("Queue is Empty")

    elif (self.rear >= self.front):

        print("Elements in the circular queue are:",end = " ")

        for i in range(self.front, self.rear + 1):

            print(self.queue[i], end = " ")

        print ()

    else:

        print ("Elements in Circular Queue are:",end = " ")

        for i in range(self.front, self.size):

            print(self.queue[i], end = " ")

        for i in range(0, self.rear + 1):

            print(self.queue[i], end = " ")

        print ()

        if ((self.rear + 1) % self.size == self.front):

            print("Queue is Full")

#__main__

ob = CircularQueue(5)

```

```
ob.enqueue(14)
ob.enqueue(22)
ob.enqueue(13)
ob.enqueue(-6)
ob.display()
print ("Deleted value = ", ob.dequeue())
print ("Deleted value = ", ob.dequeue())
ob.display()
ob.enqueue(9)
ob.enqueue(20)
ob.enqueue(5)
ob.display()
```

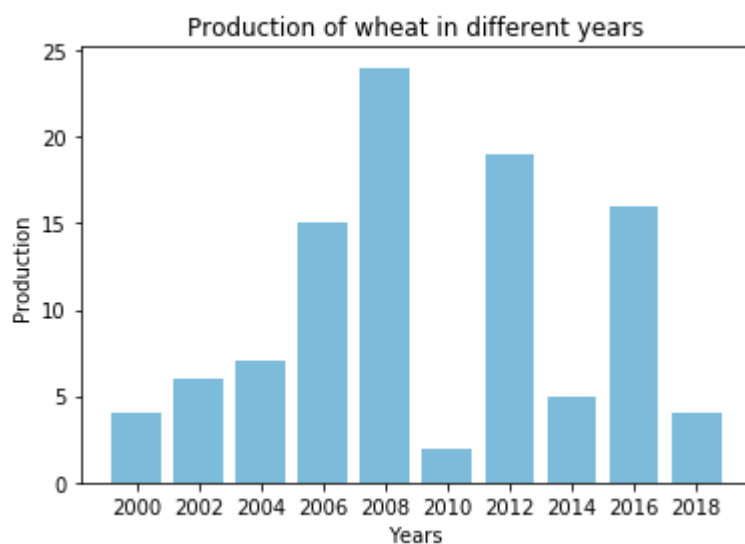
OUTPUT:

```
Elements in the circular queue are: 14 22 13 -6
Deleted value = 14
Deleted value = 22
Elements in the circular queue are: 13 -6
Elements in Circular Queue are: 13 -6 9 20 5
Queue is Full
```

#Graph based on the production of wheat in different years.

```
import matplotlib.pyplot as plt  
import numpy as np  
  
objects =  
(2000,2002,2004,2006,2008,2010,2012,2014,2016,2018)  
  
y = np.arange(len(objects))  
  
performance = [4,6,7,15,24,2,19,5,16,4]  
  
plt.bar(y, performance, align='center', alpha=0.5)  
plt.xticks(y, objects)  
plt.ylabel('Production')  
plt.xlabel('Years')  
plt.title('Production of wheat in different years')  
plt.show()
```

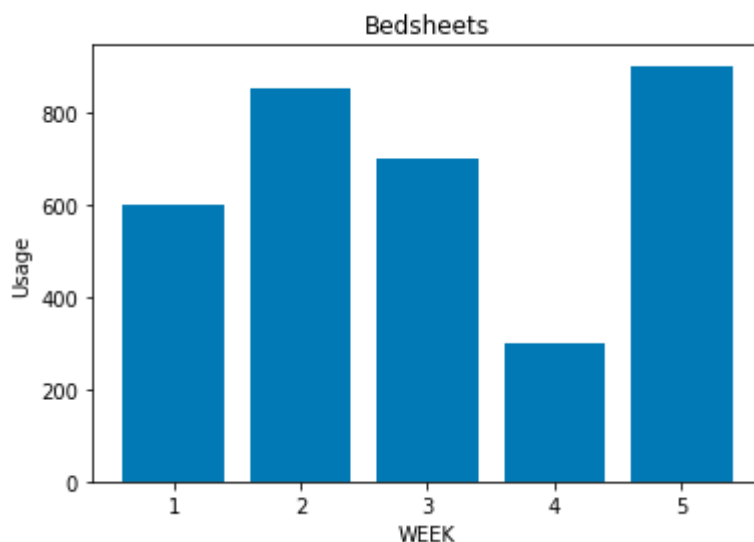
OUTPUT:



#Bar graph representing bed-sheets manufactured by a factory.

```
import matplotlib.pyplot as plt  
import numpy as np  
x=[1,2,3,4,5]  
y=[600,850,700,300,900]  
plt.bar(x,y)  
plt.xlabel("WEEK")  
plt.ylabel("Usage")  
plt.title("Bedsheets")  
plt.show()
```

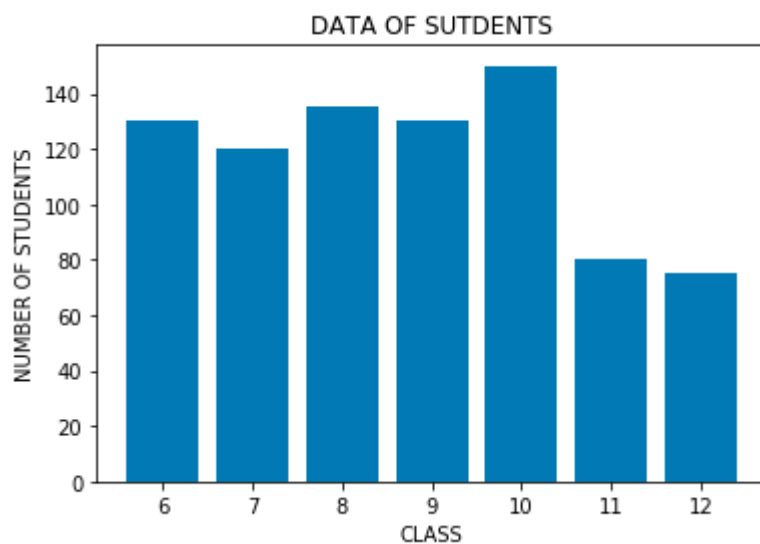
OUTPUT:



#Program for Graph of the give data.

```
import matplotlib.pyplot as plt  
import numpy as np  
a=[6,7,8,9,10,11,12]  
b=[130,120,135,130,150,80,75]  
plt.bar(a,b)  
plt.xlabel("CLASS")  
plt.ylabel("NUMBER OF STUDENTS")  
plt.title("DATA OF SUTDENTS")  
plt.show()
```

OUTPUT:

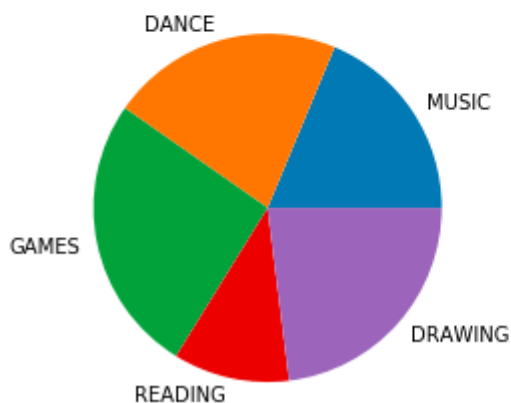


#Program to pie graph of the given data.

```
import matplotlib.pyplot as plt  
  
labels = ['MUSIC','DANCE', 'GAMES', 'READING','DRAWING']  
  
number_of_students = [130,150,180,75,160]  
  
plt.pie(number_of_students,labels=labels)
```

OUTPUT:

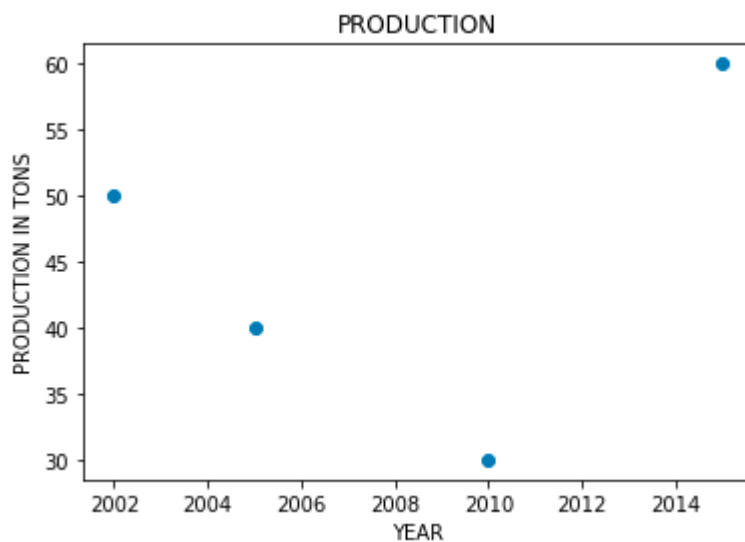
```
([<matplotlib.patches.Wedge at 0x873dc08>,  
 <matplotlib.patches.Wedge at 0x8746388>,  
 <matplotlib.patches.Wedge at 0x8746348>,  
 <matplotlib.patches.Wedge at 0x874b708>,  
 <matplotlib.patches.Wedge at 0x8752588>],  
 [Text(0.9154789352338673, 0.6098346654160165, 'MUSIC'),  
 Text(-0.30665142637607196, 1.0563923999639153, 'DANCE'),  
 Text(-1.0773212340458904, -0.22221376796643377, 'GAMES'),  
 Text(-0.23437408375699953, -1.0747412660092042, 'READING'),  
 Text(0.824628636069477, -0.7280024811593669, 'DRAWING')])
```



#Program to make a graph of the given data.

```
%matplotlib inline  
import numpy as np  
import matplotlib.pyplot as plt  
N = 500  
x = [2002,2005,2010,2015]  
y = [50,40,30,60]  
colors = (0,0,0)  
plt.title('PRODUCTION')  
plt.xlabel('YEAR')  
plt.ylabel('PRODUCTION IN TONS')  
plt.scatter(x, y)  
plt.show()
```

OUTPUT:



MYSQL

THE GIVE TABLE IS :

1:

```
mysql>
mysql> SELECT* FROM DOCTOR;
```

ID	NAME	DEPT	SEX	EXPERIENCE
101	JOHN	ENT	M	17
104	SMITH	ORTHOPEDIC	M	5
107	GEORGE	CARDIOLOGY	M	10
110	LARA	SKIN	F	3
109	K GREGORY	MEDICINE	F	9
105	JOHNSON	ORTHOPEDIC	M	10
117	LUCY	ENT	F	4
111	JILL	MEDICINE	F	12
130	BARNEY	ORTHOPEDIC	M	15

9 rows in set (0.00 sec)

2:

```
mysql> SELECT* FROM SALARY;
```

ID	BASIC	ALLOWANCE	CONSULTANCY
101	12000	1000	300
104	23000	2300	500
107	32000	4000	500
114	12000	5200	100
109	42000	1700	200
105	15900	1090	300
130	21700	2600	300

ANS

(A)

```
SELECT NAME FROM DOCTOR  
WHERE DEPT='MEDICINE'  
AND EXPERIENCE > 10;
```

```
mysql> SELECT NAME FROM DOCTOR  
-> WHERE DEPT='MEDICINE'  
-> AND EXPERIENCE > 10;  
+-----+  
| NAME |  
+-----+  
| JILL |  
+-----+  
1 row in set (0.00 sec)
```

(B)

```
SELECT AVG(BASIC+ALLOWANCE) AS 'AVERAGE_SALARY'  
FROM SALARY, DOCTOR  
WHERE SALARY.ID=DOCTOR.ID  
AND DEPT='ENT';
```

```
mysql> SELECT AVG(BASIC+ALLOWANCE) AS 'AVERAGE_SALARY' FROM SALARY, DOCTOR  
-> WHERE SALARY.ID=DOCTOR.ID  
-> AND DEPT='ENT';  
+-----+  
| AVERAGE_SALARY |  
+-----+  
| 13000.0000 |  
+-----+  
1 row in set (0.02 sec)
```

(C)

SELECT NAME, DEPT FROM DOCTOR

WHERE SEX='F'

AND EXPERIENCE > 5;

```
mysql> SELECT NAME, DEPT FROM DOCTOR
-> WHERE SEX='F'
-> AND EXPERIENCE > 5;
+-----+-----+
| NAME      | DEPT      |
+-----+-----+
| K GREGORY  | MEDICINE  |
| JILL       | MEDICINE  |
+-----+-----+
2 rows in set (0.00 sec)
```

(D)

SELECT NAME,DEPT,CONSULTANCY FROM DOCTOR AS
D,SALARY AS S

where D.SEX='M'

and d.id=s.id

and s.consultancy = (select max(consultancy) from salary);

```
mysql>
mysql> SELECT NAME,DEPT,CONSULTANCY FROM DOCTOR AS D,SALARY AS S
-> where D.SEX='M'
-> and d.id=s.id
-> and s.consultancy = (select max(consultancy) from salary);
+-----+-----+-----+
| NAME   | DEPT          | CONSULTANCY |
+-----+-----+-----+
| SMITH  | ORTHOPEDIC    | 500         |
| GEORGE | CARDIOLOGY    | 500         |
+-----+-----+-----+
2 rows in set (0.04 sec)
```

(E)

Select count(*) from doctor where sex= "f";

```
mysql> select count(*) from doctor where sex="f";
+-----+
| count(*) |
+-----+
|         4 |
+-----+
1 row in set (0.00 sec)
```

(F)

SELECT NAME FROM DOCTOR WHERE DEPT LIKE "OR%"
AND NAME LIKE "S%";

```
mysql> SELECT NAME FROM DOCTOR WHERE DEPT LIKE "OR%" AND NAME LIKE "S%";
+-----+
| NAME |
+-----+
| SMITH |
+-----+
1 row in set (0.12 sec)
```

PROGRAM 2

THE GIVEN TABLES ARE:

1:

```
mysql> SELECT * FROM STATIONARY;
```

SL_ID	StationaryName	COMPANY	PRICE
DP01	DOT PEN	ABC	10
PL02	PENCIL	XYZ	6
ER05	ERASER	XYZ	7
PL01	PENCIL	CAB	5
GP02	GEL PEN	ABC	15

5 rows in set (0.00 sec)

2:

```
mysql> SELECT * FROM CONSUMER;
```

CustId	ConsumerName	ADDRESS	S_ID
1	GOOD LEARNER	DELHI	PL01
6	WRITE WELL	MUMBAI	GP02
12	TOPPER	DELHI	DP01
15	WRITE AND DRAW	DELHI	PL02
16	MOTIVATION	BANGLORE	PL01

5 rows in set (0.00 sec)

(A)

```
SELECT* FROM CONSUMER  
WHERE ADDERSS='DELHI';
```

```
mysql> SELECT* FROM CONSUMER  
-> WHERE ADDERSS='DELHI';  
+-----+-----+-----+-----+  
| CustId | ConsumerName | ADDERSS | S_ID |  
+-----+-----+-----+-----+  
|      1 | GOOD LEARNER | DELHI   | PL01 |  
|     12 | TOPPER       | DELHI   | DP01 |  
|     15 | WRITE AND DRAW | DELHI   | PL02 |  
+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

(B)

```
SELECT*FROM STATIONARY  
WHERE PRICE <= 15 AND PRICE >= 8;
```

```
mysql> SELECT*FROM STATIONARY  
-> WHERE PRICE <=15 AND PRICE>=8;  
+-----+-----+-----+-----+  
| SL_ID | StationaryName | COMPANY | PRICE |  
+-----+-----+-----+-----+  
| DP01  | DOT PEN       | ABC    | 10    |  
| GP02  | GEL PEN       | ABC    | 15    |  
+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

(C)

```
SELECT CONSUMERNAME, ADDERSS, COMPANY, PRICE
FROM CONSUMER, STATIONARY WHERE CONSUMER.S_ID=
STATIONARY.SL_ID;
```

```
mysql> SELECT CONSUMERNAME, ADDERSS, COMPANY, PRICE FROM CONSUMER , STATIONARY
-> WHERE CONSUMER.S_ID= STATIONARY.SL_ID;
```

CONSUMERNAME	ADDERSS	COMPANY	PRICE
TOPPER	DELHI	ABC	10
WRITE AND DRAW	DELHI	XYZ	6
GOOD LEARNER	DELHI	CAB	5
MOTIVATION	BANGLORE	CAB	5
WRITE WELL	MUMBAI	ABC	15

```
5 rows in set (0.00 sec)
```

(D)

```
UPDATE STATIONARY
```

```
SET PRICE=PRICE+2;
```

```
mysql> UPDATE STATIONARY
-> SET PRICE=PRICE+2;
Query OK, 5 rows affected (0.16 sec)
Rows matched: 5 Changed: 5 Warnings: 0

mysql> SELECT* FROM STAIONARY
-> ;
ERROR 1146 (42S02): Table 'project.staionary' doesn't exist
mysql> SELECT* FROM STATIONARY
-> ;
```

SL_ID	StationaryName	COMPANY	PRICE
DP01	DOT PEN	ABC	12
PL02	PENCIL	XYZ	8
ER05	ERASER	XYZ	9
PL01	PENCIL	CAB	7
GP02	GEL PEN	ABC	17

```
5 rows in set (0.00 sec)
```


(E)

Select Distinct Address from Consumer;

```
mysql> SELECT DISTINCT ADDRSS FROM CONSUMER;  
+-----+  
| ADDRSS |  
+-----+  
| DELHI  |  
| MUMBAI |  
| BANGLORE |  
+-----+  
3 rows in set (0.02 sec)
```

(F)

Select Company, MAX(Price), MIN(Price),COUNT(*), from
Stationary Group by Company;

```
mysql> SELECT CONSUMER.CONSUMERNAME, STATIONARY.STATIONARYNAME, STATIONARY.PRICE  
-> FROM STATIONARY, CONSUMER WHERE CONSUMER.S_ID=STATIONARY.SL_ID;  
+-----+-----+-----+  
| CONSUMERNAME | STATIONARYNAME | PRICE |  
+-----+-----+-----+  
| GOOD LEARNER | PENCIL         | 7     |  
| WRITE WELL   | GEL PEN        | 17    |  
| TOPPER        | DOT PEN        | 12    |  
| WRITE AND DRAW | PENCIL        | 8     |  
| MOTIVATION    | PENCIL        | 7     |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

(G)

Select Stationary Name, price*3 From Stationary;

```
mysql> SELECT STATIONARYNAME, PRICE*3 FROM STATIONARY;  
+-----+-----+  
| STATIONARYNAME | PRICE*3 |  
+-----+-----+  
| DOT PEN        | 36     |  
| PENCIL         | 24     |  
| ERASER         | 27     |  
| PENCIL         | 21     |  
| GEL PEN        | 51     |  
+-----+-----+  
5 rows in set (0.00 sec)
```