SINDHI HIGH SCHOOL- Hebbal

COMPUTER SCIENCE

PRACTICAL FILE

2020-21

Name: Shreyas R

Class: XII

Section: A

Board Roll No:

**INDEX:**

|  |  |  |
| --- | --- | --- |
| **Sl no.** | **Chapter Name** | **Page no.** |
| 1. | Python Revision Tour 1 & 2 | 3-34 |
| 2. | Working with functions | 35-46 |
| 3. | File Handling | 47-79 |
| 4. | Data Structure | 80-92 |
| 5. | Interface Python with MYSQL | 93-103 |
| 6. | MYSQL Queries | 104-119 |

**Python Revision Tour**

**1) Write a program to input an angle in radians and display it in degrees**

r=float (input ("Enter the angle in radians : "))

d=(180/3.14)\*r

print (d,"Degrees")

**OUTPUT:**

Enter the angle in radians :15

859.8726114649681 Degrees

**2) Write a program to convert a given number of days into months and days and also display the remaining days in a month**

n=int(input("Enter the number of days : "))

m=n//30

d=n%30

print(m,"months",d,"days")

print("Remaining days :",(30-d))

**OUTPUT:**

Enter the number of days : 57

1 months 27 days

Remaining days : 3

**3) Write a program to get the 3rd side of a right angles triangle from the given two sides**

n1=int(input("Enter the 1st side :"))

n2=int(input("Enter the 2nd side :"))

n3=(n1\*\*2+n2\*\*2)\*\*0.5

print(n3)

**OUTPUT :** Enter the 1st side :5

Enter the 2nd side :4

6.4031242374328485

**4) Write a program to convert height in feet and inches to cm**

f=int(input("Enter number of feet :"))

i=float(input("Enter the number of inches : "))

f1=12\*2.5\*f

i1=2.5\*i

cm=f1+i1

print(" Height in cm ",cm)

**OUTPUT**

Enter number of feet :8

Enter the number of inches : 6

Height in cm : 255.0

**5) Write a program to convert distance in feet to inches, yards and miles**

f=float(input("Enter the number of feet :"))

i=f\*12

y=f/3

m=f/5280

print(i,"inches",y,"yards",m,"miles")

**OUTPUT**

Enter the number of feet :10

120.0 inches 3.3333333333333335 yards 0.001893939393939394 miles

**6) Write a program to convert all units of time to seconds**

n=int(input("Enter the number : "))

s=n

m=n\*60

h=n\*60\*60

d=n\*60\*60\*24

print(n,"seconds ---->",s,"seconds")

print(n,"minutes ---->",m,"seconds")

print(n,"hours ---->",h,"seconds")

print(n,"days ——>",d,"seconds")

**OUTPUT**

Enter the number : 25

25 seconds ----> 25 seconds

25 minutes ----> 1500 seconds

25 hours ----> 90000 seconds

25 days ——> 2160000 seconds

**7) Write a program to convert given seconds to day, hours, minutes and seconds**

s=int(input("Enter seconds :"))

d=s//(24\*3600)

s=s%(24\*3600)

h=s//3600

s=s%3600

m=s//60

s=s%60

print("Days",d,"hours",h,"minutes",m,"seconds",s)

**OUTPUT**

Enter seconds :4200

Days 0 hours 1 minutes 10 seconds 0

**8) write a program to find area and circumference of the circle of given diameter**

dia=int(input("Enter the diameter "))

r=dia/2

a=3.14\*r\*r

c=2\*3.14\*r

print("Area =",a,"; Circumference =",c)

**OUTPUT**

Enter the diameter 25

Area = 490.625 ; Circumference = 78.5

**9) Write a program to find the distance between 2 points**

x1=int(input("Enter the x coordinate of the 1st point "))

y1=int(input("Enter the y coordinate of the 1st point "))

x2=int(input("Enter the x coordinate of the 2nd point "))

y2=int(input("Enter the y coordinate of the 2nd point "))

distance=((x1-x2)\*\*2+(y1-y2)\*\*2)\*\*0.5

print(distance)

**OUTPUT**

Enter the x coordinate of the 1st point 3

Enter the y coordinate of the 1st point 1

Enter the x coordinate of the 2nd point 2

Enter the y coordinate of the 2nd point 6

5.0990195135927845

**10) Write a program to find the mid point of a line**

x1=int(input("Enter the x coordinate of the 1st point "))

y1=int(input("Enter the y coordinate of the 1st point "))

x2=int(input("Enter the x coordinate of the 2nd point "))

y2=int(input("Enter the y coordinate of the 2nd point "))

x=(x1+x2)/2

y=(y1+y2)/2

print(x,",",y,"is the midpoint")

**OUTPUT**

Enter the x coordinate of the 1st point 6

Enter the y coordinate of the 1st point 4

Enter the x coordinate of the 2nd point 2

Enter the y coordinate of the 2nd point 3

4.0 , 3.5 is the midpoint

**11) Write a program to get the descending order of 3 input numbers**

x=int(input("Enter the 1st number "))

y=int(input("Enter the 2nd number "))

z=int(input("Enter the 3rd number "))

print("THE DESCENDING ORDER IS")

if x>=y and x>=z:

if y>z:

print(x,y,z)

else:

print(x,z,y)

elif y>=x and y>=z:

if x>z:

print(y,x,z)

else:

print(y,z,x)

elif z>=x and z>=y:

if x>y:

print(z,x,y)

else:

print(z,y,x)

**OUTPUT**

Enter the 1st number 6

Enter the 2nd number 1

Enter the 3rd number 9

THE DESCENDING ORDER IS

9 6 1

**12) Write a program to perform the input operation with the input numbers**

n1=int(input("Enter 1st number "))

n2=int(input("Enter 2nd number "))

c=input("Enter the operator ")

a=0

if c=='+':

a=n1+n2

if c=='i':

a=n1-n2

if c=='\*':

a=n1\*n2

if c=='/':

a=n1/n2

if c=="//":

a=(n1//n2)

if c=='%':

a=n1%n2

print(n1,c,n2,"=",a)

**OUTPUT**

Enter 1st number 5

Enter 2nd number 26

Enter the operator \*

5 \* 26 = 130

**13) Write a program to check if a character is a digit, lowercase , uppercase or special character**

c=input("Enter the character ")

if c>='a' and c<='z':

print("LOWERCASE")

elif c>='A' and c<='Z':

print("UPPERCASE")

elif c>='0' and c<='9':

print("DIGIT")

else:

print("SPECIAL CHARACTERS ")

**OUTPUT**

Enter the character: #

SPECIAL CHARACTERS

**14) Write a program to find the solutions of a quadratic equation**

a=int(input("Enter the coefficient of x^2 : "))

b=int(input("Enter the coeffeicient of x :"))

c=int(input("Enter the constant :"))

x1=(-b+((b\*\*2-4\*a\*c)\*\*0.5))/(2\*a)

x2=(-b-((b\*\*2-4\*a\*c)\*\*0.5))/(2\*a)

print("The roots are",x1," , ",x2)

**OUTPUT**

Enter the coefficient of x^2 : 1

Enter the coeffeicient of x :-5

Enter the constant :6

The roots are 3.0 , 2.0

**15) Write a program to check if the input number is 1, 2, 3 digit or other than this**

n=int(input("Enter the number "))

if n>999:

print("Invalid input ")

else:

if n>99:

print("3 digit")

elif n>9:

print("2 digit")

elif n>=0:

print("1 digit")

else:

print("Invalid input ")

**OUTPUT**

Enter the number 456

3 digit

**16) Write a program to print the cube of numbers from 15 to 20**

for i in range (15,21):

print(i\*i\*i)

**OUTPUT**

3375

4096

4913

5832

6859

8000

**17) Write a program to print the square root of nos from 1 to 50**

for i in range(1,51):

print(i\*\*0.5)

**OUTPUT**

1.0

1.4142135623730951

1.7320508075688772

2.0

2.23606797749979

2.449489742783178

2.6457513110645907

2.8284271247461903

3.0

3.1622776601683795

3.3166247903554

3.4641016151377544

3.605551275463989

3.7416573867739413

3.872983346207417

4.0

4.123105625617661

4.242640687119285

4.358898943540674

4.47213595499958

4.58257569495584

4.69041575982343

4.795831523312719

4.898979485566356

5.0

5.0990195135927845

5.196152422706632

5.291502622129181

5.385164807134504

5.477225575051661

5.5677643628300215

5.656854249492381

5.744562646538029

5.830951894845301

5.916079783099616

6.0

6.082762530298219

6.164414002968976

6.244997998398398

6.324555320336759

6.4031242374328485

6.48074069840786

6.557438524302

6.6332495807108

6.708203932499369

6.782329983125268

6.855654600401044

6.928203230275509

7.0

7.0710678118654755

**18) Write a program to convert seconds to minutes and seconds**

n=int(input("Enter number of seconds : "))

m=n//60

s=n%60

print(m,"minutes",s,"seconds")

**OUTPUT**

Enter number of seconds : 125

2 minutes 5 seconds

**19) Write a program to keep inputing numbers into a list and print it when "done" is typed in**

sum=0

while True:

ch=input("Enter the character :")

if ch=="Done " or ch=="done":

break

else:

b=float(ch)

sum+=b

print("Sum :",int(sum))

**OUTPUT**

Enter the character :6

Enter the character :2

Enter the character :7

Enter the character :done

Sum : 15

**20) Write a program to print this table**

**1 2 3 4 5 6 7 8 9 10**

**2 4 6 8 10 12 14 16 18 20**

**3 6 9 12 15 18 21 24 27 30**

**4 8 12 16 20 24 28 32 36 40**

**5 10 15 20 25 30 35 40 45 50**

**6 12 18 24 30 36 42 48 54 60**

**7 14 21 28 35 42 49 56 63 70**

**8 16 24 32 40 48 56 64 72 80**

**9 18 27 36 45 54 63 72 81 90**

**10 20 30 40 50 60 70 80 90 100**

for i in range (1,11):

for j in range (1,11):

print(i\*j,end="\t")

print()

**OUTPUT**

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100

**21) WAP That reads an integer n from the keyboard, computes and displays the sum of numbers from n to 2n, if n is non negative. If n is a negative no then the sum of numbers from 2n to n**

n=int(input("Enter the value :"))

sum=0

if n<0:

x=-(n)

for i in range(x,x\*2):

sum+=i

else:

for i in range(2\*n,n,-1):

sum+=i

print("The sum of numbers :",sum)

**Output**

Enter the value :3

The sum of numbers : 15

**22) Write a program to print Fibonacci numbers up to the given no of terms**

a=0

b=1

n=int(input("Enter number of terms : "))

print(a,end=",")

print(b,end=",")

for i in range (0,n-2):

c=a+b

a,b=b,c

print(c,end=",")

**OUTPUT**

Enter number of terms : 5

0,1,1,2,3,

**23) Write a program to check whether the input number is armstrong**

n=input("Enter the number :")

m=0

s=0

for i in range (0,len(n)):

m=int(n[i])

s+=(m\*\*3)

if s==int(n):

print("It is an armstrong number")

else:

print("It is not an armstrong number “)

**OUTPUT**

Enter the number :153

It is an armstrong number

**24) Write a program to check whther a string is a palindrome**

s=input("Enter the string : ")

l=len(s)

if l%2==0:

mid=int((l+1)/2)

else:

mid=int(l/2)

rev=-1

t=True

for a in range(mid):

if s[a]==s[rev]:

a+=1

rev-=1

else:

t=False

print("Not a palindrome ")

break

if t==True:

print("It is a palindrome”)

**OUTPUT**

Enter the string: MAM

It is a palindrome

**25) Write a program to print the following patterns**

**i) 5 4 3 2 1**

**4 3 2 1**

**3 2 1**

**2 1**

**1**

**ii) A**

**A B**

**A B C**

**A B C D**

**iii). \***

**\* \* \***

**\* \* \* \* \***

**iv) 1^2+2^2+3^2+......+ n ^2 =**

**v) (x^2)/4 - (x^3)/6 + (x^4)/8 - ……**

**vi) 1+(1+2)+(1+2+3)+.....upto n terms =**

i)for i in range(5,0,-1):

for j in range (i,0,-1):

print(j,end="")

print()

**OUTPUT**

**5 4 3 2 1**

**4 3 2 1**

**3 2 1**

**2 1**

**1**

ii) for i in range(1,6):

for j in range(5,i,-1):

print("",end="")

for k in range (1,i):

print(chr(k+64),end="")

print()

**OUTPUT**

**A**

**A B**

**A B C**

**A B C D**

iii)

for i in range (1,4):

for j in range (3,i,-1):

print("",end="")

for k in range(1,(i\*2)):

print("\*",end="")

print()

**OUTPUT**

\*

\* \* \*

\* \* \* \* \*

iv) S=0

n=int(input("Enter the number "))

for i in range (1,n+1):

S+=(i\*\*2)

print(“1^2+2^2+3^2+......+",n,"^2=",S)

**OUTPUT**

Enter the number 5

**1^2+2^2+3^2+......+ 5 ^2= 55**

v)

x=int(input("Enter x "))

n=int(input("Enter number of terms "))

s=x

for i in range (2,n+1):

if i%2==0:

s+=(x\*\*i)/(i\*\*2)

else:

s-=(x\*\*i)/(i\*\*2)

print(s)

**OUTPUT**

Enter x 6

Enter number of terms 4

72.0

vi)

s=0

j=0

n=int(input("Enter the number "))

for i in range (1,n+1):

s+=i

j+=s

print("1+(1+2)+(1+2+3)+.....",n,"terms = “,j)

**OUTPUT**

Enter the number 5

1+(1+2)+(1+2+3)+..... 5 terms = 35

**26) Write a program to print out the number of lowercase, uppercase, digits and alphabets**

s=input("Enter the string ")

n1=n2=n3=n4=0

for i in s:

if i.islower():

n1+=1

if i.isupper():

n2+=1

if i.isalpha():

n3+=1

if i.isdigit():

n4+=1

print("LOWER",n1)

print("UPPER",n2)

print("ALPHABET",n3)

print("DIGIT",n4)

**OUTPUT**

Enter the string "HEllo, WHATSuP"

LOWER 4

UPPER 8

ALPHABET 12

DIGIT 0

**27) Write a program to get a string made os the 1st 2 and last 2 characters of a string**

s=input("Enter the string ")

s1=""

l=len(s)

s1+=s[0:2]

s1+=s[l-2:l]

print(s1)

**OUTPUT**

Enter the string HELLOO

HEOO

**28) Write a program to create a new string where every 1st character of word is converted to "$"**

s=input("Enter the string ")

s1="$"

l=len(s)

i=1

while i<l:

if s[i]=="":

s1+=" $"

i+=2

else:

s1+=s[i]

i+=1

print(s1)

**OUTPUT**

Enter the string "what is this"

$hat $s $his

**29) Write a program to do as said if the length of a string is atleast 3, then add "ing" if it already ends with "ing" then add "ly".if its less then leave it unchanged**

s=input("Enter the string ")

l=len(s)

if l<3:

print(s)

else:

if s[l-3:l]=="ing":

s+="ly"

print(s)

else:

s+="ing"

print(s)

**OUTPUT**

Enter the string "doing"

doingly

**30) Write a program to create a string by replacing the first and last characters of a string**

s=input("Enter the string ")

s1=""

l=len(s)

for i in range (0,l):

if i==0:

s1+=s[l-1]

elif i==l-1:

s1+=s[0]

else:

s1+=s[i]

print(s1)

**OUTPUT**

Enter the string "WHAT"

THAW

**31) Write a program to reverse a stirng if length is not more than 4**

s=input("Enter the string ")

l=len(s)

if l<=4:

s1=""

for i in range (l-1,-1,-1):

s1+=s[i]

print(s1)

else:

print(s)

**OUTPUT**

Enter the string "DOG"

GOD

**32) Write a program to if a string contains at least 2 uppercase in its first 4 letters, then convert the whole string to uppercase**

s=input("Enter the string ")

l=len(s)

c=0

for i in range (0,4):

if s[i].isupper():

c+=1

if c>=2:

s=s.upper()

print(s)

**OUTPUT**

Enter the string "HEllo"

HELLO

**33) Write a program to capitlize the first letter of each word in a string**

s=input("Enter the string ")

l=len(s)

s1=s[0].upper()

i=1

while i<l:

if s[i]=="":

s1+=""

s1+=s[i+1].upper()

i+=2

else:

s1+=s[i]

i+=1

print(s1)

**OUTPUT**

Enter the string "What is your name"

What Is Your Name

**34) Write a program to**

**i) repeatedly prompt for a string or 'q' to quit**

**ii) convert lower case to upper case and vice-versa. leave all others unaffected**

c = True

while c:

s = input("Enter the string ")

if s != "q" and s != "Q":

s1 = ""

l = len(s)

for i in range(0, l):

if s[i].isupper():

s1 += s[i].lower()

elif s[i].islower():

s1 += s[i].upper()

else:

s1 += s[i]

print(s1)

else:

c = False

**OUTPUT**

Enter the string "HELLO, MY NAME IS SHREYAS"

hello, my name is shreyas

Enter the string "Q"

**35) Write a Program to Input an int and a string. from the str input any digit if present and add it with the int input.**

n=int(input("Enter the number "))

s=input("Enter the string ")

s1=""

l=len(s)

for i in range (0,l):

if s[i].isdigit():

s1+=s[i]

n1=int(s1)

n2=n1+n

print(n, "+",n1,"=",n2)

**OUTPUT**

Enter the number 34

Enter the string "HELLO 5"

34 + 5 = 39

**36) Write a program to return the longest string of a list with its size**

l=eval(input("Enter the list "))

s=l[0]

m=len(s)

for i in l:

j=len(i)

if j>m:

k=i

m=j

print(k,m)

**OUTPUT**

Enter the list ["HELLO","ARE","WATERING"]

WATERING 8

**37) Write a program to print the number of occurences of a number in a list**

l=eval(input("Enter the list "))

n=int(input("Enter the number "))

c=0

for i in range (0,len(l)):

if l[i]==n:

c+=1

print(c)

**OUTPUT**

Enter the list [1,2,4,2,5,2,3]

Entre the number 2

3

**38) Write a program to print the input list in descendig order**

l=eval(input("Enter the list "))

for j in range (0,len(l)):

while l[j]>l[j-1] and j>0:

l[j],l[j-1]=l[j-1],l[j]

j-=1

print(l)

**OUTPUT**

Enter the list [2,5,1,7.3]

[7.3, 5, 2, 1]

**39) Write a program to find all numbers in the range which are perfect squares and sum of all digits is less than 10**

a=int(input("Enter the start of range :"))

b=int(input("Enter the end of range :"))

lst=[]

lst1=[]

for i in range(a,b+1):

if (i\*\*0.5)% 1==0 :

lst.append(i)

for j in lst:

sum=0

ele=j

while j>0:

digit=j%10

j=int(j/10)

sum+=digit

if sum<=10:

lst1.append(ele)

print( "List of number in range that are perfect squares and sum of digits less than 10 : ")

print(lst1)

**OUTPUT**

Enter the start of range :1

Enter the end of range :50

List of number in range that are perfect squares and sum of digits less than 10 :

[1, 4, 9, 16, 25, 36]

**40) Write a program that read a list of words and returns the length of the longest word**

lst=eval(input("Enter a list of words :"))

length=0

for i in lst:

if len(i)>length:

length=len(i)

print("The length of the longest word in the list is :",length)

**OUTPUT**

Enter a list of words :["HE","MANGO","SHREYAS"]

The length of the longest word in the list is : 7

**41) Write a program to find frequency of each item of a tuple**

tup=eval(input("Enter a tuple :"))

print("Element \t\t Frequency")

print()

dupl=[]

for i in tup:

if i not in dupl:

ele=i

dupl.append(i)

count=0

for j in tup :

if j==ele:

count+=1

print(ele,"\t\t",count)

**OUTPUT**

Enter a tuple :(1,4,5,6,10,3,"ARE",1,2)

Element Frequency

1 2

4 1

5 1

6 1

10 1

3 1

ARE 1

2 1

**42) Write a program to find the hcf and lcm of 2 numbers**

a=int(input("Enter the 1st number "))

b=int(input("Enter the 2nd number "))

if a>b:

small=b

elif b>a:

small=a

for i in range(1,small+1):

if (a%i==0) and (b%i==0):

hcf=i

lcm=(a\*b)/hcf

print("LCM IS",lcm)

print("HCF IS",hcf)

**OUTPUT**

Enter the 1st number 5

Enter the 2nd number 12

LCM IS 60.0

HCF IS 1

**43) Write a program to input a tuple and finding the second largest element of the tuple**

t=eval(input("Enter a tuple of numbers :"))

large=max(t)

length=len(t)

slarge=0

for i in range(length):

if slarge < t[i]<large:

slarge=t[i]

print("Second largest element :",slarge)

**Output**

Enter a tuple of numbers :(6,4,10,9,1,2)

Second largest element : 9

**44) Write a program that finds an element’s index position in a tuple without using index()**

t=eval(input("Enter a tuple :"))

char=input("Enter the character to search :")

if char in t:

count=0

for i in t:

if i!=char:

count+=1

else:

break

print(char,"is at index",count)

else:

print(char,"is not in ",t)

**Output:**

Enter a tuple :(1,3,4,"a",7,1,8)

Enter the character to search :a

a is at index 3

**45) Write a program that returns a tuple containing x, x^2,x^3 and x^4. Read five integers from the user and for each integer read, print that value raised to the powers 2,3 and 4.**

for n in range(5):

x=int(input("Enter number :"))

t=(x,x\*\*2,x\*\*3,x\*\*4)

print(x,"raised to power 1,2,3,4 :=",t)

**Output**

Enter number :1

1 raised to power 1,2,3,4 := (1, 1, 1, 1)

Enter number :2

2 raised to power 1,2,3,4 := (2, 4, 8, 16)

Enter number :3

3 raised to power 1,2,3,4 := (3, 9, 27, 81)

Enter number :4

4 raised to power 1,2,3,4 := (4, 16, 64, 256)

Enter number :5

5 raised to power 1,2,3,4 := (5, 25, 125, 625)

**46) Write a program to make a dictionary of names of winners as keys and no of medals**

medals=dict()

n=int(input("No of winners :"))

i=0

while i<n:

a=input("Enter name :")

b=int(input("Enter medals"))

medals[a]=b

i+=1

print("The dictionary is : ")

print(medals)

**Output:**

No of winners :3

Enter name :shreyas

Enter medals3

Enter name :anonymous1

Enter medals2

Enter name :anonymous2

Enter medals4

The dictionary is :

{'shreyas': 3, 'anonymous1': 2, 'anonymous2': 4}

**47) Write a program to input n names and numbers in a dictionary and to search the number**

phone=dict()

n=int(input("No of entries :"))

i=0

while i<n:

a=input("Enter name :")

b=int(input("Enter number"))

phone[a]=b

i+=1

l=phone.keys()

x=input("enter name to be searched ")

for i in l:

if i==x:

print(x,phone[i])

break

else:

print("Entry not found")

**Output**

No of entries :2

Enter name :shreyas

Enter number1234567

Enter name :anony

Enter number7654321

enter name to be searched shreyas

shreyas 1234567

**48) Write a program to input names of the country , its capital and currency and to display the same in tabular format**

country=dict()

n=int(input("No of entries :"))

i=0

while i<n:

c=input("Enter country name :")

cap=input("Enter capital name :")

cur=input("Enter currency name :")

country[c]=(cap,cur)

i+=1

for i in country:

print("\n")

print(i,end=" ")

for j in country[i]:

print(j,end=" ")

print()

**Output**

No of entries :2

Enter country name :India

Enter capital name :delhi

Enter currency name :rupee

Enter country name :USA

Enter capital name :washington

Enter currency name :dollar

India delhi rupee

USA washington dollar

**49) Write a program that checks for the presence of a value inside a dictionary and prints its key**

info={'riya':'cs','mark':'eco','sihpreet':'eng','kamal':'env'}

a=input("Enter the value to be searched for :")

if a in info.values():

for i in info:

if info[i]==a:

print("The key of the given value is",i)

break

else:

print("Given value does not exist")

**Output**

Enter the value to be searched for :cs

The key of the given value is riya

**50) let my\_points={'a':(4,3),'b':(1,2),'c':(5,1)}….. Write a program to calculate the maximum value from within all the values tuples at same index. Eg. Maximum for 0th index will be computed from 4,1,5 – all the entries at 0th index in the value tuple.**

my\_points={'a':(4,3),'b':(1,2),'c':(5,1)}

high=[0,0]

a=0

for i in range(2):

a=0

for j in my\_points.keys():

val=my\_points[j][i]

if a==0:

high[i]=val

a+=1

if val>high[i]:

high[i]=val;

print("Max value at index(my\_points,",i,")=",high[i])

**Output**

Max value at index(my\_points, 0 )= 5

Max value at index(my\_points, 1 )= 3

**51) Write a program to sort using insertion sort**

l=eval(input("Enter a list :"))

print("Original list is :",l)

for i in range(1,len(l)):

key=l[i]

j=i-1

while j>=0 and key<l[j]:

l[j+1]=l[j]

j=j-1

else:

l[j+1]=key

print("List after sorting :",l)

**Output**

Enter a list :[6,5,2,3,1,4,7]

Original list is : [6, 5, 2, 3, 1, 4, 7]

List after sorting : [1, 2, 3, 4, 5, 6, 7]

**52) Write a program to sort using bubble sort**

l=eval(input("Enter a list :"))

print("Original list is :",l)

for i in range(len(l)):

for j in range(0,len(l)-i-1):

if l[j]>l[j+1]:

l[j],l[j+1]=l[j+1],l[j]

print(" list after sorting :",l)

**Output**

Enter a list :[3,7,1,6,4,2,5]

Original list is : [3, 7, 1, 6, 4, 2, 5]

list after sorting : [1, 2, 3, 4, 5, 6, 7]

**WORKING WITH FUNCTIONS**

**1) Write a program to accept the P T R from the user and write a function program to display the simple interest**

def si(p,r,t):

i=(p\*r\*t)/100

return i

P=int(input("Enter the principal "))

R=int(input("Enter the rate "))

T=int(input("Enter the Time "))

I=si(P,R,T)

print("Simple Interest =“,I)

**OUTPUT**

Enter the principal 200

Enter the rate 5

Enter the Time 1

SIMPLE INTEREST= 10.0

**2) Write a program to check if the number entered is even or odd**

def odd(x):

if x%2==0:

i=0

else:

i=1

return i

n=int(input("Enter the number to be checked "))

h=odd(n)

if n==0:

print("IT IS EVEN ")

else:

print("IT IS ODD”)

**OUTPUT**

Enter the number to be checked 3

IT IS ODD

**3) Write a program to find the sum of the series 1!/1+2!/2+3!/3+4!/4+5!/5 using the function.**

def fact(x):

p=1

for i in range (1,x+1):

p=i\*p

a=p/x

return a

S=0

for i in range(1,6):

S+=fact(i)

print("1!/1+2!/2+3!/3+4!/4+5!/5 =",S)

**OUTPUT**

1!/1+2!/2+3!/3+4!/4+5!/5 = 34.0

**4) Write a program to check whether a number is a prime number or not**

def prime(x):

n=0

for i in range(1,x+1):

if x%i==0:

n+=1

if n==2:

k=1

else:

k=0

return k

N=int(input("ENTER THE NUMBER TO BE CHECKED "))

h=prime(N)

if h==0:

print("THE NUMBER ISNT PRIME ")

else:

print("THE NUMBER IS PRIME “)

**OUTPUT**

ENTER THE NUMBER TO BE CHECKED 9

THE NUMBER ISNT PRIME

**5) Write a program to find the Max of three numbers**

def max(x,y,z):

if x>=y and x>=z:

return x

elif y>=x and y>=z:

return y

elif z>=x and z>=y:

return z

a=int(input("Enter the 1st number "))

b=int(input("Enter the 2nd number "))

c=int(input("Enter the 3rd number "))

h=max(a,b,c)

print("The max= “,h)

**OUTPUT**

Enter the 1st number 3

Enter the 2nd number 4

Enter the 3rd number 5

The max= 5

**6) Write a program to sum all the numbers in a list**

def sum(l):

s=0

for i in l:

s+=i

return s

lis=eval(input("Enter the list "))

h=sum(lis)

print("The sum of all elements in the list is”,h)

**OUTPUT**

Enter the list [8,3,3,4]

The sum of all elements in the list is 18

**7) Write a program to function to multiply all the numbers in a list**

def prod(l):

p=1

for i in l:

p\*=i

return p

lis=eval(input("Enter the list "))

h=prod(lis)

print("The product of all elements in the list is”,h)

**OUTPUT**

Enter the list [2,3,4]

The product of all elements in the list is 24

**8) Write a program to check whether a number is in a given range**

def range(l,u,n):

p=0

if n>=l and n<=u:

p=1

return p

up=int(input("Enter the upper limit "))

lo=int(input("Enter the lower limit "))

no=int(input("Enter the number to be checked "))

s=range(lo,up,no)

if s==1:

print("It lies in the range")

else:

print("It doesnt lie in the range")

**OUTPUT**

Enter the upper limit 9

Enter the lower limit 2

Enter the number to be checked 6

It lies in the range

**9) Write a program that accepts a string and calculate the number of upper case letters and lower case letters.**

def strin(s):

u=0

l=0

for i in s:

if i.isupper():

u+=1

elif i.islower():

l+=1

return u,l

S=input("Enter the string")

U,L=strin(S)

print("No of uppercase",U)

print("No of lowercase”,L)

**OUTPUT**

Enter the string"ShReYAs"

No of uppercase 4

No of lowercase 3

**10) Write a program to print the even numbers from a given list.**

def even(l):

e=[]

for i in l:

if i%2==0:

e.append(i)

return e

L=eval(input("Enter the list "))

h=[]

h=even(L)

for i in h:

print(i)

**OUTPUT**

Enter the list [8,1,6,3,4]

8

6

4

**11) Write a program to find the roots of a quadratic equation**

def quad(a,b,c):

D=(b\*\*2)-4\*a\*c

x1=(-b-D)/2\*a

x2=(-b+D)/2\*a

return x1,x2

A=int(input("Enter the coefficient of x^2 "))

B=int(input("Enter the coefficient of x "))

C=int(input("Enter the constant term "))

r1,r2=quad(A,B,C)

print("The roots of the quadratic are”,r1,”and",r2)

**OUTPUT**

Enter the coefficient of x^2 1

Enter the coefficient of x -5

Enter the constant term 6

The roots of the quadratic are 2.0 and 3.0

**12) Write a function program that calculates the arithmetic mean of list elements. List is passed as an argument to the function**

def am(l):

s=0

for i in l:

s+=i

a=s/len(l)

return a

lis=eval(input("Enter the list "))

print("The arithmetic mean of the list is “,am(lis))

**OUTPUT**

Enter the list [5,2,3,10]

The arithmetic mean of the list is 5.0

**13) Write a function program to find factorial of a number**

def fact(n):

p=1

for i in range (1,n+1):

p\*=i

return p

a=int(input("Enter the number to be calculated "))

print("The factorial of the number is”,fact(a))

**OUTPUT**

Enter the number to be calculated 5

The factorial of the number is 60

**14) Write a function program to calculate sum of first n natural numbers**

def SUM(n):

s=0

for i in range(1,n+1):

s+=i

return s

a=int(input("Enter the number to be calculated "))

print("The sum of all natural numbers until”,a,"is",SUM(a))

**OUTPUT**

Enter the number to be calculated 5

The sum of all natural numbers until 5 is 15

**15) Write a function program to display Fibonacci series**

def fibo(n):

a=1

b=1

print(a,",",b,end=", ")

for i in range(3,n+1):

c=a+b

print(c,end=", ")

a,b=b,c

a=int(input("Enter the number of terms in the series "))

fibo(a)

**OUTPUT**

Enter the number of terms in the series 4

1 , 1, 2,3

**16) Write a function to calculate volume of a box with length, width and height of the box passed as parameters**

def vol(l,b,h):

v=l\*b\*h

return v

L=int(input("Enter the Length "))

B=int(input("Enter the Breadth "))

H=int(input("Enter the Height "))

V=vol(L,B,H)

print("Volume of the box is”,V)

**OUTPUT**

Enter the Length 5

Enter the Breadth 4

Enter the Height 8

Volume of the box is 160

**17) Write a void function that receives a 4 digit number and calculates the sum of the squares of each digit in the number**

def digit(n):

s=0

while n>0:

t=n%10

s+=(t\*\*3)

n=n//10

print(s)

N=int(input("Enter the value "))

digit(N)

**OUTPUT**

Enter the value 2103

14

**18) Write a definition of a method(function) odd-sum() which receives the list of numbers as parameter and finds the sum of odd numbers in the list**

def odd\_sum(l):

s=0

for i in l:

if i%2==1:

s+=i

return s

L=eval(input("Enter the list "))

S=odd\_sum(L)

print(S)

**OUTPUT**

Enter the list [1,3,5,6,7]

15

**19) Write a method in Python to find and display the prime numbers between 2 to N where N is passed as a parameter to the function.**

def prime(n):

for i in range(2,n+1):

s=0

for j in range(1,i+1):

if i%j==0:

s+=1

if s==2:

print(i)

n=int(input("Enter the number "))

prime(n)

**OUTPUT**

Enter the number 10

2

3

5

7

**20) Write a function that takes a list that is sorted in ascending order and a number as argument. The function should do the following:**

**1) Insert the number passed as argument in a sorted list**

**2) Delete the number from the list**

def add(l,n):

l.append(n)

return l

def remove(l,n):

l.pop(n)

return l

s=[]

L=eval(input("Enter the list in ascending order "))

N=int(input("Enter the number "))

c=int(input("Enter 1 to add or 2 to remove "))

if c==1:

s=add(L,N)

elif c==2:

s=remove(L,N)

print(s)

**OUTPUT**

Enter the list in ascending order [1,3,5,7,9]

Enter the number 3

Enter 1 to add or 2 to remove 2

[1, 5, 7, 9]

**21) Write a function that receives two tuples and creates a third that contains all the elements of the first followed by all elements of the second.**

def tup(t1,t2):

s=[]

for i in t1:

s.append(i)

for i in t2:

s.append(i)

h=tuple(s)

return h

n=eval(input("Enter the first tuple "))

m=eval(input("Enter the second tuple "))

h=tup(n,m)

print(h)

**OUTPUT**

Enter the first tuple (2,3,4)

Enter the second tuple (6,2,1)

( 2, 3, 4, 6, 2,1)

**22) Define a function overlapping that takes in two lists and returns true if they have at least one member in common, false otherwise**

def comm(l1,l2):

t=False

if t==False:

for i in l1:

for j in l2:

if i==j:

t=True

break

return t

l1=eval(input("Enter the list "))

l2=eval(input("Enter the list "))

s=comm(l1,l2)

print(s)

**OUTPUT**

Enter the list [1,2,3,4]

Enter the list [4,5,6,7]

True

**FILE HANDLING**

**1) Write a program to display all the records in a file along with line or record number.**

f=open("poem.txt",'r')

s=""

c=1

while s:

s=f.readline()

print(s,end="")

print("Line ",c)

c+=1

f.close()

**OUTPUT**

1 If you could read my mind,

2 Youâ€™d see a thousand papers

3 Filled with broken poetries

4 And deadbeat proses

5 Full of woeful verses

6 With mournful pieces

7 Of unfinished stories

8 That are yet to be written

9 And failed to be spoken;

**2) A text file contains alphanumeric text. Write a program that reads this text file and prints only the numbers or digits from the file.**

f=open("poem.txt",'r')

s=""

c=0

while s:

s=f.readline()

for i in s:

if i.isnumeric():

c+=1

print(c)

f.close()

**OUTPUT**

0

**3) Write a function to count the number of lines in a text file poem.txt which is starting with alphabet** ‘**A**’

f=open("poem.txt",'r')

s=f.readlines()

c=0

for i in s:

if i[0]=="A" or i[0]=="a":

c+=1

print(c)

**OUTPUT**

2

**4) A text file answer.txt contains the text as shown below. Write a program to remove all the lines that contain the character** ‘**a**’ **in this file and write other lines into another file**

f=open("Answer.txt",'r')

s=f.readlines()

l=[]

l1=[]

for i in s:

c=1

if 'a' in i:

l.append(i)

else:

l1.append(i)

h=open("answer1.txt",'w')

h.writelines(l1)

h=open("answer1.txt",'r')

print(h.read())

g=open("answer2.txt",'w')

g=open("answer2.txt",'r')

print(g.read())

**OUTPUT**

We know this will never occur.

Hoe mediocre our world would be without this single most powerful letter.

Letter 'a' is a wonderful letter.

It is impossible to think of a sentence without it.

**5) Write a program to increase the salary by rupees 2,000 of the employee having emp number as 1251 in the file emp1.dat**

import pickle

f=open("emp1.bin",'wb+')

n=int(input("Enter the number of employees"))

for i in range (n):

emp={}

no=int(input("Enter the emp no "))

sal=int(input("Enter the salary "))

emp['NUMBER']=no

emp['SALARY']=sal

pickle.dump(emp,f)

f.close()

f=open("emp1.bin",'rb+')

emp1={}

try:

while True:

emp1=pickle.load(f)

if emp1['NUMBER']==1251:

emp1['SALARY']+=2000

print(emp1)

except EOFError:

f.close()

**OUTPUT**

Enter the number of employees2

Enter the emp no 1251

Enter the salary 2000

Enter the emp no 1231

Enter the salary 1000

{'NUMBER': 1251, 'SALARY': 4000}

{'NUMBER': 1231, 'SALARY': 1000}

**6) Write a function in python to read lines from a text file india.tv, to find and display the occurrence of the word India**

f=open("india.tv",'r')

s=""

c=0

l="India"

while s:

s=f.readline()

if l in s:

c+=1

print(c)

**OUTPUT**

4

**7) Write a method in python to read lines from a text file “Poem.txt” and display those lines, which are starting with an alphabet k**

f=open("poem.txt",'r')

s=f.readlines()

for i in s:

if i[0]=='k':

print(i)

**OUTPUT**

 \_\_\_\_\_\_\_ NO OUTPUT\_\_\_\_\_\_\_

**8) Consider the following definition of dictionary MULTIPLEX, write a method in python to search and display all the content in a pickled file cinema.dat, where mtype key of the dictionary is matching with the value** ‘**comedy**’

**MULTIPLEX = {**‘**MNO**’**:\_\_\_\_\_\_\_,**’**MNAME**’**:\_\_\_\_\_\_\_\_,**’**MTYPE**’**:\_\_\_\_\_\_\_\_\_}**

import pickle

f=open('search','wb')

n=int(input("Enter the number of records "))

DICTIONARY={}

for i in range(n):

l=[]

no=input("Enter the movie number ")

na=input("Enter the movie name ")

ty=input("Enter the movie type ")

DICTIONARY["MN0"]=no

DICTIONARY["MNAME"]=na

DICTIONARY["MTYPE"]=ty

pickle.dump(DICTIONARY,f)

f.close()

f=open('search','rb')

e={}

try:

while True:

emp=pickle.load(f)

if emp["MTYPE"]=="comedy":

print(emp)

except EOFError:

f.close()

**OUTPUT**

Enter the number of records 2

Enter the movie number 1

Enter the movie name ti

Enter the movie type comedy

Enter the movie number 2

Enter the movie name ta

Enter the movie type horror

{'MN0': '1', 'MNAME': 'ti', 'MTYPE': ‘comedy'}

**9) Take a sample text file and find the most commonly occurring word. Also, list the frequencies of words in the text file**

f=open("india.tv",'r')

s=f.read()

d={}

w=s.split()

for i in w:

if i in d:

d[i]+=1

else:

d[i]=1

for i in d:

print(i,d[i])

**OUTPUT**

India 4

is 4

the 4

fastest 1

growing 1

economy 1

looking 2

for 2

more 1

investments 1

around 1

globe 1

The 1

whole 1

world 1

at 1

as 1

a 1

great 1

market 1

Most 1

of 2

Indians 1

can 1

see 1

heights 1

that 1

capable 1

reaching 1

**10) write a program that read a text file and display the count of lowercase and uppercase letters in the file note(Note: use above text file answer.txt)**

f=open("Answer.txt",'r')

s=f.read()

c1=0

c2=0

for i in s:

if i.islower():

c1+=1

elif i.isupper():

c2+=1

print("lower case",c1)

print("Upper case",c2)

**OUTPUT**

lower case 147

Upper case 4

**11) Write a function stats() that accepts a filename and reports the file**’**s longest word**

def stats(n):

f=open(n,'r')

s=f.read()

l=s.split()

m=l[0]

M=len(m)

for i in range(1,len(l)):

h=l[i]

if len(h)>M:

m=h

M=len(m)

print(m,M)

s=input("Enter file name")

stats(s)

**OUTPUT**

Enter file namePoem.txt

unfinished 10

**12) Create a file text2.txt with all the words that dont start from vowel from text1.txt and read the file**

def VOWELSWORD(n1,n2):

f=open(n1,'r')

s=f.read()

l=[]

s1=""

for i in s:

if i=="" or i==s[-1]:

l.append(s1+' ')

s1=""

continue

else:

s1+=i

f.close()

f=open(n2,'w')

s=['A','E','I','O','U']

for i in l:

if i[0] in s:

continue

else:

f.write(i+'\n')

f.close()

f=open(n2,'r')

print(f.read())

print(l)

m1=input("Enter the file to be referred to")

m2=input("Enter the file to be printed to")

VOWELSWORD(m1,m2)

**OUTPUT**

Enter the file to be referred toTEXT1.txt

Enter the file to be printed toTEXT2.txt

Caary

When

Rain

['Caary ', 'Umbrella ', 'And ', 'Overcoat ', 'When ', 'It ', 'Rain ']

**13) Print the number of uppercase in text file “article.txt”**

f=open("ARTICLE.txt",'r')

s=f.read()

c=0

for i in s:

if i.isupper():

c+=1

print("NUMBER OF UPPERCASE =“,c)

**OUTPUT**

NUMBER OF UPPERCASE = 33

**14) Write a function to read notes.txt and print the longest and the shortest words**

f=open("NOTES.txt",'r')

s=f.read()

l=s.split()

def max(l):

M = l[0]

m = len(M)

for i in l:

if len(i)>m:

M=i

m=len(i)

print("Longest word: ",M)

def min(l):

M = l[0]

m = len(M)

for i in l:

if len(i) < m:

M = i

m = len(i)

print("Shortest Word: ",M)

max(l)

min(l)

**OUTPUT**

"Longest word: "communication

"Shortest word: "A

**15) WAF to read a file and print the number of a’s and e’s**

def count(l):

f=open(l,'r')

s=f.read()

c1=0

c2=0

l1=['A','a']

l2=['E','e']

for i in s:

if i in l1:

c1+=1

elif i in l2:

c2+=1

print("Number of A's =",c1)

print("Number of E's =",c2)

count("Poem.txt")

**OUTPUT**

Number of A's = 11

Number of E's = 25

**16) WAF to reverse the string of a list “input.txt”**

def reverse(l):

f=open(l,'r')

s=f.read()

le=len(s)

s1=''

for i in range((le-1#),-1,-1):

s1+=s[i]

print(s1)

reverse("input.txt")

**OUTPUT**

YRTNUOC YM SI AIDNI

**17) WAF to count the number of words starting with a capital letter**

def count(l):

f=open(l,'r')

s=f.read()

l=s.split()

c=0

for i in l:

if i[0].isupper():

c+=1

print("NUMBER OF UPPER CASE ",c)

count("coordinate.txt")

**OUTPUT**

NUMBER OF UPPER CASE 25

**18) WAF to count the number of “do”s in “memo.txt”**

def count(l):

f=open(l)

s=f.read()

l=s.split()

c=0

l1=["DO","do"]

for i in l:

if i in l1:

c+=1

print(c)

count("MEMO.TXT")

**OUTPUT**

2

**19) WAF to print the words that start and end with the same letter**

def wordcount(l):

f=open(l,'r')

s=f.read()

l=s.split()

for i in l:

if i[0]==i[-1]:

print(i)

wordcount(“ARTICLE.txt")

**OUTPUT**

that

a

a

a

**20) WAF to copy a file “report.txt” into a file “finerep.txt” with all its content in lowercase except for the first character of the file and the character after every fullstop.**

def copy(l,n):

f=open(l,'r')

s=f.read()

s=s.lower()

le=len(s)

s1=""

i=0

while i<le:

if i==0:

s1+=s[i].upper()

i+=1

elif s[i]==".":

s1+=". "+s[i+2].upper()

i+=3

else:

s1+=s[i]

i+=1

f.close()

f=open(n,'w')

f.write(s1)

copy("Report.Txt","finerep.txt")

**OUTPUT**

 \_\_\_\_\_\_\_ NO OUTPUT\_\_\_\_\_\_\_

**21) WAF to copy a file into another, except to replace 2 consecutive blank spaces with just one.**

def copy(n,l):

f=open(n,'r')

s=f.read()

le=len(s)

s1=""

i=0

while i<le:

if s[i]=="" and s[i+1]=="":

s1+=""

i+=2

else:

s1+=s[i]

i+=1

f.close()

f=open(l,'w')

f.write(s1)

copy("Report.txt","copy1.txt")

**OUTPUT**

 \_\_\_\_\_\_\_ NO OUTPUT\_\_\_\_\_\_\_

**22)WAF to count the number of uppercase in “article.txt"**

def upper(n):

f=open(n,'r')

s=f.read()

c=0

for i in s:

if i.isupper():

c+=1

return c

k=upper("Article.txt")

print("NO OF UPPERCASE =“,k)

**OUTPUT**

NO OF UPPERCASE = 33

**23) WAP read a file "tele.txt" and print its content in 2 columns**

f=open("tele.txt",'r')

s=f.read()

l=s.split()

le=len(l)

i=0

while i<le:

print(l[i],end='\t')

print(l[i+1])

i+=2

**OUTPUT**

KUMAR 81358031

SACHIN 43242197

KEVIN 63683686

**24) WAP to merge two text file into one text file**

n=input("Enter the name of the new file ")

n1=input("Enter the 1st old file ")

n2=input("Enter the 2nd old file ")

f1=open(n1,'r')

s1=f1.read()

f2=open(n2,'r')

s2=f2.read()

f=open(n,'w')

s3=s1+s2

f.write(s3)

**OUTPUT**

Enter the name of the new file 123.txt

Enter the 1st old file Poem.txt

Enter the 2nd old file baba.txt

**25) WAP to count the number of words which are smaller than 4 characters**

def DISPLAY\_WORDS(n):

f=open(n,'r')

s=f.read()

l=s.split()

for i in l:

if len(i)<4:

print(i)

DISPLAY\_WORDS("story.TXT")

**OUTPUT**

I

do

it,

if

you

me

to

do

it.

It

**26) WAP to print a file until ‘$’ appears**

f=open("dollar.txt",'r')

s=f.read()

for i in s:

if i=="$":

break

else:

print(i,end='')

**OUTPUT**

In WW1, there was one day, when Hitler was so badly injured,

that he would just die if there was no medical aid provided. It was then the British Private

**27) WAP to define a function ‘fileout’ to keep writing lines into a file, as long as the user wants**

def fileout(n):

f=open(n,'w')

while True:

c=input("Enter the line to be added or 'q' to quit ")

if c!='q':

f.write(c)

f.write("\n")

else:

break

fileout("user.txt")

**OUTPUT**

Enter the line to be added or 'q' to quit "HELLO"

Enter the line to be added or 'q' to quit "ANONE THERE"

Enter the line to be added or 'q' to quit I WILL LEAVE YOU HERE"

Enter the line to be added or 'q' to quit "q"

**28) A file sports.dat contains information in following format : Event ^ Participant. Write a function that would read contents from the file and crate a file atheletic.dat copying only those records from sports.dat where the even is “Athletics”.**

f=open("sports.dat",'r')

s=f.readlines()

m=[]

for i in s:

j=i.split()

if j[0].lower()=="athletics":

print(i)

m.append(i)

f.close()

f=open("athletic.dat",'w+')

f.writelines(m)

f.read()

**OUTPUT**

**athletics Shreyas**

**athletics Bolt**

**29) A file contains a list of telephone numbers in the following form : Name Number The names contain only one word. The names and tekephone numbers are separated by white spaces. Write a program to read a file and display its contents in two columns.**

f=open("tele.dat",'r')

print("NAME","%15s" % "NUMBER")

s=f.readlines()

for i in s:

k=i.split()

print(k[0],"%15s" % k[1])

**OUTPUT**

KUMAR 81358031

SACHIN 43242197

KEVIN 63683686

**30) Write a program that appends the contents of one file to another. Have the program take the file names from the user.**

n1=input("Enter the name of the original file ")

n2=input("Enter the name of the file where to be copied ")

while True:

try:

f=open(n1,'r')

s=f.read()

g = open(n2, 'w')

g.write(s)

break

except:

print("check the file names")

break

**OUTPUT ————-**

Enter the name of the original file info.dat

Enter the name of the file where to be copied newone.txt

**31) Write a program with function DISPLAY WORDS () to read lines from a text file story.TXT and display those words which are less than 4 characters.**

def display\_words(n):

m=[]

f=open(n,'r')

s=f.read()

g=s.split()

for i in g:

if len(i)<4:

m.append(i)

print(m)

n=input("Enter the file to be read ")

display\_words(n)

**OUTPUT**

Enter the file to be read story.txt

['In', 'was', 'one', 'was', 'so', 'he', 'die', 'if', 'was', 'no', 'aid', 'It', 'was', 'the', 'had', 'to', 'by.', 'Was', 'he', 'a', 'to', 'his', 'he', 'at', 'the', 'at', 'let', 'him', 'too', 'die', 'on', 'his', 'But', 'he', 'to', 'get', 'to', 'the', 'So', 'had', 'he', 'not', 'a', 'a', 'no', 'war', 'USA', 'nad', 'the', 'and', 'the', 'was', 'to', 'do', 'the', 'end', 'of']

**32) Write a program that counts the number of characters up to the first S in a text file.**

def count(n):

c=0

f=open(n,'r')

s=f.read()

n=len(s)

for i in range(n):

if s[i]=="$":

print(i+1)

break

elif i==(n-1):

print("NOT FOUND")

n=input("Enter the name to be checked ")

count(n)

**OUTPUT**

Enter the name to be checked story.txt

61

**33) Write a program that will create an object called fileout for writing, associate it with the file name STRS.txt. The code should keep on writing strings to it as long as the user wants.**

class fileout():

def write(self):

h='n'

f=open("nice2.txt",'w')

i=""

while i!=h:

i = input("Enter after each line and type 'n' to stop ")

if i!=h:

f.write(i+'\n')

h=fileout()

h.write()

**OUTPUT**

Enter after each line and type 'n' to stop my

Enter after each line and type 'n' to stop name is

Enter after each line and type 'n' to stop Shreyas

Enter after each line and type 'n' to stop n

**34) Consider the following definition of dictionary member, write a method in Python to write the content in a pickled file member.dat**

**Member={‘MemberNo’: \_\_\_\_\_\_\_\_,’Name’:\_\_\_\_\_\_\_\_\_\_\_)**

import pickle

member1={}

def member():

file=open("members.dat",'a+b')

n=int(input("Input number of entries :"))

for i in range(n):

no=int(input("Input member number :"))

name=input("Input member name :")

member1[no]=name

pickle.dump(member1,file)

file.close()

def update():

with open('members.dat', 'ab') as f:

dicti = {}

n = input('Enter member number : ')

name = input("Enter member name : ")

dicti[n] = name

member1.update(dicti)

pickle.dump(eval(str(member1)), f)

def read\_file():

data = ' '

with open('members.dat', 'rb') as file:

try:

while data:

data = eval(str(pickle.load(file)))

for i, j in data.items():

print(i, j)

except EOFError:

file.close()

while True:

print('1 . TO ADD DATA 2 . TO UPDATE 3. TO READ')

ch = input('Enter your choice : ')

if ch == '1':

member()

break

elif ch == '2':

update()

break

elif ch=='3':

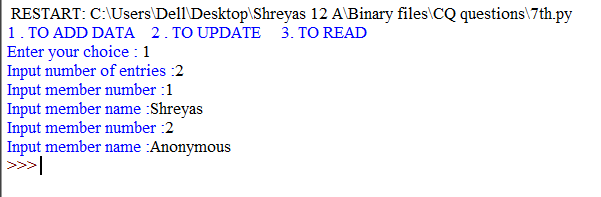
read\_file()

else:

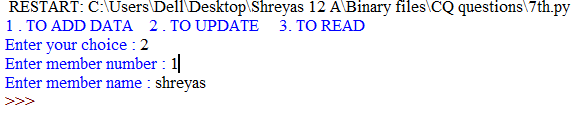
print("Wrong Input Retry")

continue

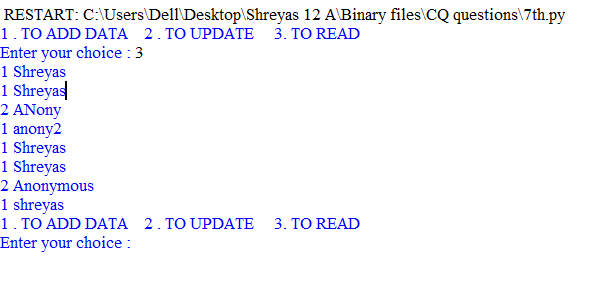
**OUTPUT 1**

****

**OUTPUT 2**



**OUTPUT 3**

**35)Consider the foll definition of dictionary Staff, write a method in Python to search and display the content In a pickled file staff dot where staff code key of the dictionary is matching wrth ‘S0105'Staff = eStaffoode' : :Name' :**

import pickle

file=open("staff.dat",'a+b')

staff={}

n=int(input("Input number of entries :"))

for i in range(n):

code="Staff code:" + input("Enter the staff code :")

name="Staff name:" + input("Enter the staff name :")

staff[code]=name

pickle.dump(staff,file)

file.seek(0,0)

data=pickle.load(file)

print("%5s"%"Staff code","%20s"%"Staff Name")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

for i in data:

print("%5s"%i[11:],"%25s"% data[i][5:])

file.close()

file1=open("Staff.dat",'rb')

search=input("Enter the code that u want to search :")

data=pickle.load(file1)

for i in data:

if i[11:] == search:

print("Record found ")

print("%10s"%"Staff code","%20s"%"Staff Name")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("%10s"%i[11:],"%20s"% data[i][5:])

break

else:

print("Record not found ")

file1.close()

**OUTPUT**

Input number of entries :1

Enter the staff code :2

Enter the staff name :anony2

Staff code Staff Name

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 name:anony2

Enter the code that u want to search :1

Record not found

**36) Considering the following definition of dictionary COMPANY, write a method in Python to search and dispLay the content in a pickled file compony.daf, where CompID key of the dictionary Is matching with the va4ue '1005' Company = {'ComplY - ;cname, Turnover'-=**

import pickle

def write():

ans = 'y'

m = []

while ans.lower() == 'y':

f = open("company.dat", 'wb+')

cid = input("Enter the comp ID : ")

cna = input("Enter C NAME :")

tu=input("Turnover")

company = {}

company["COMPANY ID"] = cid

company["C NAME"] = cna

company["TURNOVER"]=tu

m.append(company)

ans = input("Continue ?")

pickle.dump(m, f)

f.close()

f = open("company.dat", ‘rb+')

emp = []

while True:

try:

emp = pickle.load(f)

except EOFError:

break

print("%15s" % "C ID ", "%20s" % "C NAME ", "%20s" % "TURNOVER")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

for i in emp:

j = list(i.values())

print()

if j[0] == "1005":

print("%15s" % j[0], "%20s" % j[1], "%20s" % j[2])

write()

**OUTPUT**

Enter the comp: ID 1

Enter C NAME: Apple

Turnover : 200

Continue ?N

C ID C NAME TURNOVER

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1 Apple 200

**37) Write a function To search and display details of at trains whose destination Is "Delhi from a binary file "Train.dar. Assuming the binary file is containing the objects of the following dictionary type**

**Train= {Tno:\_\_\_\_\_\_\_,From:\_\_\_\_\_\_\_,To:\_\_\_\_\_\_\_\_\_\_\_)**

import pickle

dict={}

file=open("train.dat",'ab')

n=int(input("Enter the no of entries : "))

for i in range(n):

no=int(input("Enter train no :"))

from\_place=input("From which place : ")

to\_place=input("To which place : ")

dict['Tno:']=no

dict['From:']=from\_place

dict['To:']=to\_place

pickle.dump(dict,file)

file.close()

file=open("train.dat",'rb')

data=pickle.load(file)

print(data)

**OUTPUT**

Enter the no of entries : 2

Enter train no :1

From which place : Bengaluru

To which place : delhi

Enter train no :2

From which place : chennai

To which place : mumbai

{'Tno:': 1, 'From:': 'Bengaluru', 'To:': 'delhi'}

**38) Write a Python program to read a given CSV file having a tab delimiter**

import csv

with open('csvfile2.csv', 'r') as f:

data = csv.reader(f,delimiter = '\t')

for i in data:

print(','.join(i))

**OUTPUT**

a b c

e f g

h i j

**39) Write a -function that reads a CSV file and creates another CSV file with the same content but with a different delimiter**

import csv

with open('csvfile2.csv', mode='w') as file:

csvwriter = csv.writer(file, delimiter=' ')

csvwriter.writerow(['a', 'b', 'c'])

csvwriter.writerow(['e', 'f', 'g'])

csvwriter.writerow(['h', 'i', 'j'])

def rewrite():

aList = []

file = open('csvfile2.csv', "r")

reader = csv.reader(file, delimiter=',')

# next(reader, None) # Skip the header but I want to preserve it in the output csv file

for row in reader:

for col in row:

aList.append(col.lower())

file.close()

file = open('csvfile2', 'w')

csvwriter = csv.writer(file, delimiter=',')

for a in aList:

csvwriter.writerow(a.split())

rewrite()

**OUTPUT**

['a b c']

['e f g']

['h i j']

**40) Write a function that reads a CSV file and updates another CSV file with the name content except the lines beginning with 'check**

import csv

with open('csvfile3.csv', mode='w') as file:

csvwriter = csv.writer(file, delimiter=' ')

csvwriter.writerow(['a', 'b', 'c'])

csvwriter.writerow(['e', 'f', 'g'])

csvwriter.writerow(['h', 'i', 'j'])

def rewrite():

aList = []

file = open('csvfile3.csv', "r")

reader = csv.reader(file, delimiter=',')

# next(reader, None) # Skip the header but I want to preserve it in the output csv file

for row in reader:

for col in row:

aList.append(col.lower())

print(aList)

for i in range(len(aList)):

if aList[i].split()[0] == 'check':

aList.pop(i)

file.close()

file = open('csvfile3', 'w')

csvwriter = csv.writer(file, delimiter=',')

for a in aList:

csvwriter.writerow(a.split())

rewrite()

**OUTPUT:** NO OUTPUT

**41) Write a program to convert a string to binary form**

msg="HELLO !"

f=open('myfile.bin','wb')

f.write(msg.encode())

f.write(b'Anyone there ???')

f.close()

**OUTPUT**

NO OUTPUT

**42) Write a program to read binary in string**

f=open('myfile.bin','rb')

s=f.read()

print(s)

s=s.decode()

print(s)

**OUTPUT**

b'HELLO !Anyone there ???'

HELLO !Anyone there ???

**43) Write a Program to create binary file and store few records in it.**

s=20

with open("NAMES.dat",'wb') as f:

ans='y'

while ans.lower()=='y':

name=input("Enter name ")

l=len(name)

name=name+(s-l)\*' '

name=name.encode()

f.write(name)

ans=input("y/n? ")

**OUTPUT**

Enter name Shreyas

y/n? N

**44) Write a menu driven to create, read and update a file using the encoding concept**

import pickle

def create\_file():

f=open("accnt.dat",'wb')

t = "No transaction"

am=0

b = 1000

emp=[]

N=int(input("Enter the number of records "))

for i in range (0,N):

acc=input("Enter acc number ")

n=input("Enter name ")

b=int(input("Enter current balance "))

emp.append([acc,n,t,am,b])

print("\*\*\*\*\*\*\*\*")

pickle.dump(emp,f)

def read\_file():

emp=[]

f=open("accnt.dat",'rb')

print("%10s" % "ACC NO", "%20s" % "NAME", "%20s" % "TRANSACTION TYPE","%10s" % "TRANSACTION AMOUNT","%10s" % "CURRENT BALANCE")

while True:

try:

emp=pickle.load(f)

except EOFError:

break

for e in emp:

print("%10s" % e[0], "%20s" % e[1], "%20s" % e[2],"%10s" % e[3],"%10s" % e[4])

def update\_file():

us=input("Enter the acc no ")

f = open("accnt.dat", 'rb')

while True:

try:

emp = pickle.load(f)

except EOFError:

break

nemp = []

h = False

for e in emp:

if e[0]==us:

h=True

c=input("ENTER THE TRANSACTION TYPE ")

if c.lower() == "credit":

e[4] = e[4] + e[3]

nemp.append(e)

elif c.lower == "debit":

if c <= 1000:

print("TRANSACTION NOT POSSIBLE")

else:

e[4] = e[4] - e[3]

nemp.append(e)

else:

nemp.append(e)

if h==False:

print("USER NOT VALID ")

g=open("newp1.dat",'wb')

pickle.dump(nemp,g)

g.close()

f=open("accnt.dat",'wb')

g=open("newp1.dat",'rb')

while True:

try:

emp=pickle.load(g)

except EOFError:

break

pickle.dump(emp,f)

ans = 'y'

while ans.lower()=='y':

print("1-create file")

print("2-read file")

print("3-update file")

c = int(input("Enter your choice "))

if c==1:

create\_file()

if c==2:

read\_file()

if c==3:

update\_file()

ans=input("CONTINUE ? ")

**OUTPUT**

1-create file

2-read file

3-update file

Enter your choice 1

Enter the number of records 2

Enter acc number 1

Enter name Shreyas

Enter current balance 500

\*\*\*\*\*\*\*\*

Enter acc number 2

Enter name Anony

Enter current balance 100

\*\*\*\*\*\*\*\*

CONTINUE ?Y

1-create file

2-read file

3-update file

Enter your choice 2

ACC NO NAME TRANSACTION TYPE TRANSACTION AMOUNT CURRENT BALANCE

1 Shreyas No transaction 0 500

2 Anony No transaction 0 100

CONTINUE ?Y

1-create file

2-read file

3-update file

Enter your choice 3

Enter the acc no 2

ENTER THE TRANSACTION TYPE CREDIT

CONTINUE ?N

45**) Write a Program to access record randomly**

s=20

n=int(input("Enter the record number "))

with open("NAMES.dat",'rb') as f:

f.seek(s\*(n-1))

k=f.read(s)

if len(k)==0:

print("no record")

else:

print(k.decode())

**OUTPUT**

Enter the record number 3

no record

**DATA STRUCTURES**

**1) QUEUE MENU PROGRAM**

def Empty(Q):

if len(Q)==0:

return True

else:

return False

def Enqueue(Q,item):

Q.append(item)

if len(Q)==1:

front=rear=0

else:

rear=len(Q)-1

def Dequeue(Q):

if Empty(Q):

return "Underflow"

else:

val=Q.pop(0)

if len(Q)==0:

front=rear=None

return val

def Peek(Q):

if Empty(Q):

return "Underflow"

else:

front=0

return Q[front]

def Show(Q):

if Empty(Q):

print("Queue is empty \n\n\n")

else:

t=len(Q)-1

print("From FRONT :=",end=' ')

front=0

i=front

rear =len(Q)-1

while (i<=rear):

print(Q[i],"<==",end=' ')

i+=1

print()

print("\n\n\n")

Q=[]

front=rear=None

while True:

print("\*\*\*\*\*\*\*\*\* QUEUE IMPLEMENTATION \*\*\*\*\*\*\*\*\*")

print("1.ENQUEUE")

print("2.DEQUEUE")

print("3.PEEK")

print("4.SHOW")

print("0.EXIT")

ch=int(input("Enter your choice: "))

if ch==1:

val=int(input("Enter item to insert"))

Enqueue(Q,val)

elif ch==2:

val=Dequeue(Q)

if val=="Underflow":

print("Queue is empty")

print("\n")

else:

print("Deleted item was :",val)

print("\n")

elif ch==3:

val=Peek(Q)

if val=="Underflow":

print("Queue is empty \n\n\n")

else:

print("Topmost item is :", val)

print("\n")

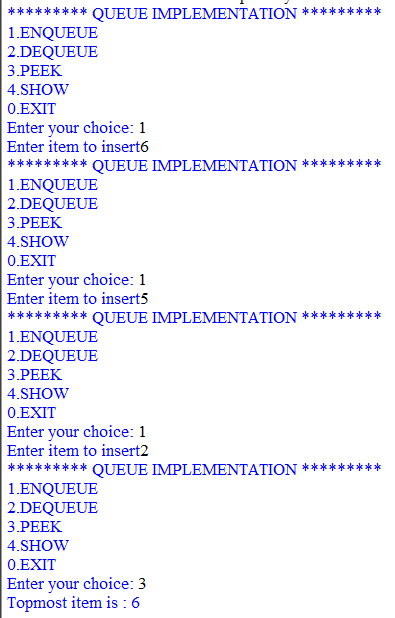
elif ch==4:

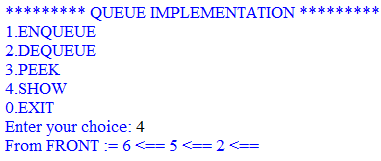
Show(Q)

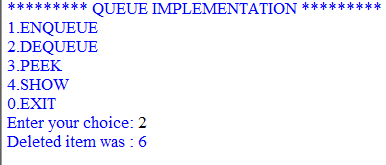
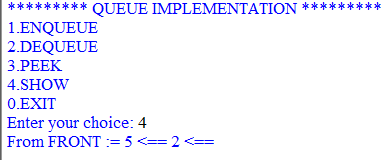
elif ch==0:

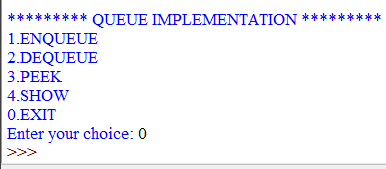
break

**OUTPUT -**









**2) STACK MENU PROGRAM**

def empty(s):

if len(s)==0:

return True

else:

return False

def Push(l,k):

l.append(k)

top=len(s)-1

def pop(s):

if empty(s):

return "Underflow"

else:

v=s.pop()

if len(s)==0:

top= None

else:

top=len(s)-1

return v

def Peek(s):

if empty(s):

return "Underflow"

else:

top=len(s)-1

return s[top]

def Show(s):

if empty(s):

print("Stack is empty. Nothing to delete \n\n\n")

else:

top=len(s)-1

print("From Top :=",end=' ')

while (top>=0):

print(s[top],"<=",end=' ')

top-=1

print()

print("\n\n\n")

s=[]

top=None

while True:

print(" \*\*\*\*\*\*\*\*\*\*STACK IMPLEMETATIONS \*\*\*\*\*\*\*\*\*\*\*")

print("1.PUSH")

print("2.POP")

print("3.PEEK")

print("4.DISPLAY")

print("0.EXIT")

ch=int(input("Enter your choice "))

if ch==1:

vl=int(input("Enter item to insert "))

print("\n\n\n")

Push(s,vl)

elif ch==2:

val=pop(s)

if val =="Underflow":

print("Stack is empty")

print("\n\n\n")

else:

print("Deleted Item :",val)

print("\n\n\n")

elif ch==3:

val=Peek(s)

if val=="Uderfolw":

print("Stack is empty")

else:

print("Topmost item is :",val)

print("\n\n\n")

elif ch==4:

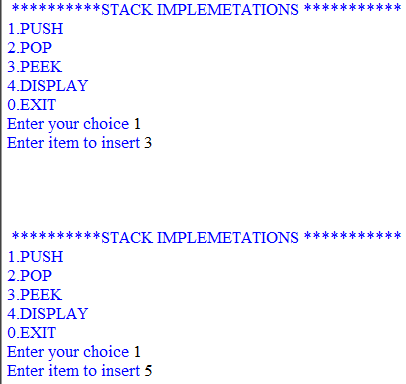
Show(s)

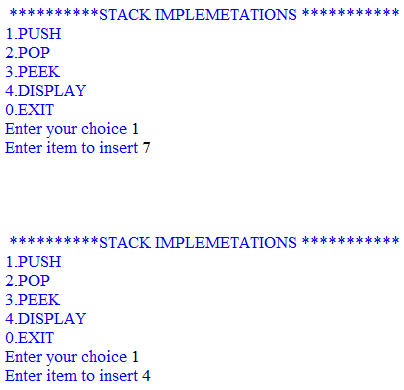
elif ch==0:

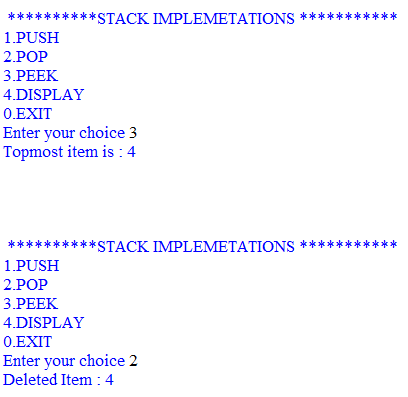
print("Bye")

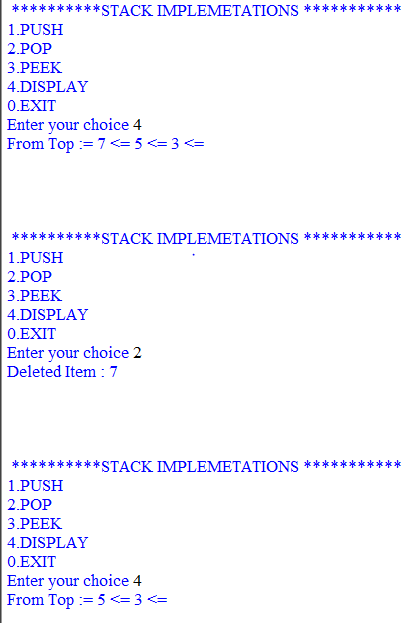
break

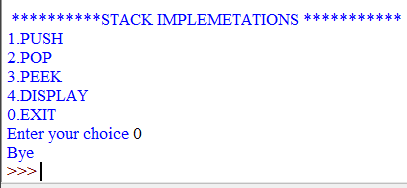
**OUTPUT**











3) **Write a program to search for an element using linear search**

def search(l,k):

le=len(l)

for i in range(le):

if l[i]==k:

return i

return None

d=[]

n=int(input("Enter the number of terms "))

for i in range(n):

s=int(input("Enter the item "))

d.append(s)

k=int(input("Enter the number to be searched "))

j=search(d,k)

if j!=None:

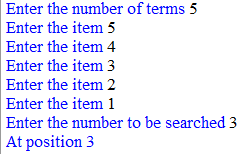
a=j+1

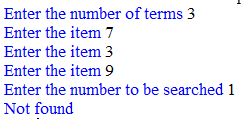
print("At position",a)

else:

print("Not found ")

**OUTPUT**





4)**Write a program to search for an element using binary search**

def search(l,k):

low=0

high=len(l)-1

while low<=high:

mid=(low+high)//2

if l[mid]==k:

return mid

elif l[mid]>k:

high=mid-1

else:

low=mid+1

return -1

d=[]

n=int(input("Enter the number of terms "))

for i in range(n):

s=int(input("Enter the item "))

d.append(s)

k=int(input("Enter the number to be searched "))

j=search(d,k)

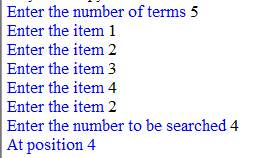
if j!= None :

print("At position",(j+1))

else:

print("Not found ")

**OUTPUT**

****

5) **Write a program to insert into a sorted list**

import bisect

l=[70,60,50,40,30]

l.reverse()

print(l)

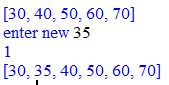
k=int(input("enter new "))

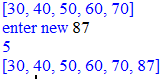
print(bisect.bisect(l,k))

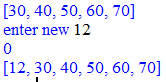
bisect.insort(l,k)

print(l)

**OUTPUT**







6)  **2 dimensional lists**

r=int(input("Enter the no of rows "))

c=int(input("Enter the no fo columns "))

mat=[]

for i in range(r):

R=[]

for j in range(c):

n=int(input("Enter the value to be stored at index ["+str(i)+","+str(j)+"]"))

R.append(n)

mat.append(R)

for i in range(r):

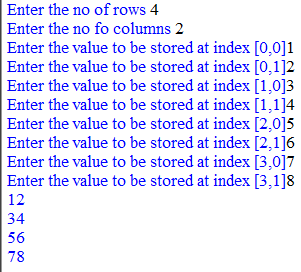
for j in range (c):

print(mat[i][j],end="")

print()

print()

**OUTPUT**



**Interface Python with MYSQL**

**1) Write a program to establish a connection with mysql**

import sys

import mysql.connector as ms

con=ms.connect(host=’localhost’,user=’root’,passwd=’tiger’,database=’class12’)

if con.is\_connected():

print(‘Connection has been made’)

else:

print(‘not connected’)

**OUTPUT**

Connection has been made

**2) Write a program to Create a table students**

import mysql.connector as ms

import sys

con=ms.connect(host='localhost',user='root',passwd='tiger',database='mysql')

cur=con.cursor()

print("Welcome to student database")

cur.execute("DROP TABLE IF EXISTS std")

cur.execute("CREATE TABLE std (rollno decimal(3,0),name varchar(10), class decimal(2,0), sex varchar(4))")

con.commit()

**OUTPUT**

Welcome to student database

**3) Write a program to display details of employees whose salary is greater than a given values and a given department no**

import sys

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

a=int(input("enter sal"))

n=int(input("enter deptno"))

x="select \* from emp where sal >{} and deptno = {}".format(a,n)

cur.execute(x)

data= cur.fetchall()

if data !=None:

for row in data:

print(\*row,sep=" : ")

else:

print("no record forund")

**OUTPUT**

enter sal950

enter deptno20

7566 : JONES : MANAGER : 7839 : 1981-04-02 : 2975.00 : None : 20

7788 : SCOTT : ANALYST : 7566 : 1982-12-09 : 3000.00 : None : 20

7876 : ADAMS : CLERK : 7788 : 1983-01-12 : 1100.00 : None : 20

**4) Write a program that displays the details of employees with department no 30 and job is salesman**

import sys

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

cur.execute("select \* from emp where deptno='30' and job='SALESMAN' ")

data=cur.fetchall()

for row in data:

print(\*row, sep=' : ')

**OUTPUT**

7499 : ALLEN : SALESMAN : 7698 : 1981-02-20 : 1600.00 : 300.00 : 30

7521 : WARD : SALESMAN : 7698 : 1981-02-22 : 1250.00 : 500.00 : 30

7654 : MARTIN : SALESMAN : 7698 : 1981-09-28 : 1250.00 : 1400.00 : 30

7844 : TURNER : SALESMAN : 7698 : 1981-09-08 : 1500.00 : 0.00 : 30

**5) Write a Program that displays the details of employees whose name starts from M**

import sys

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

cur.execute("select \* from emp where ename like 'M%' ")

data=cur.fetchall()

for row in data:

print(\*row, sep=' : ')

**OUTPUT**

7654 : MARTIN : SALESMAN : 7698 : 1981-09-28 : 1250.00 : 1400.00 : 30

7934 : MILLER : CLERK : 7782 : 1982-01-23 : 1300.00 : None : 10

**6) Write a Program to print details of employees whose salary is greater than 2500 in descending order of name**

import sys

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

cur.execute("select \* from emp where sal>2500 order by ename desc ")

data=cur.fetchall()

for row in data:

print(\*row, sep=' : ')

**OUTPUT**

7788 : SCOTT : ANALYST : 7566 : 1982-12-09 : 3000.00 : None : 20

7839 : KING : PRESIDENT : None : 1981-11-17 : 5000.00 : None : 10

7566 : JONES : MANAGER : 7839 : 1981-04-02 : 2975.00 : None : 20

7902 : FORD : ANALYST : 7566 : 1981-12-03 : 3000.00 : None : 20

7698 : BLAKE : MANAGER : 7839 : 1981-05-01 : 2850.00 : None : 30

**7) Write a Program to display the details of employees whose salary is greater than the users input**

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

n=int(input('enter sal'))

cur.execute('select \* from emp where sal>'+str(n))

data=cur.fetchall()

rec=cur.rowcount

if data != None:

for i in data:

print(\*i,sep=':')

else:

print('no record found')

**OUTPUT**

enter sal1500

7499:ALLEN:SALESMAN:7698:1981-02-20:1600.00:300.00:30

7566:JONES:MANAGER:7839:1981-04-02:2975.00:None:20

7698:BLAKE:MANAGER:7839:1981-05-01:2850.00:None:30

7782:CLARK:MANAGER:7839:1981-06-09:2450.00:None:10

7788:SCOTT:ANALYST:7566:1982-12-09:3000.00:None:20

7839:KING:PRESIDENT:None:1981-11-17:5000.00:None:10

**8) Write a Program to display all contents of the emp table**

import mysql.connector as ms

con=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=con.cursor()

cur.execute('select \* from emp')

data = cur.fetchall()

if data != None:

for i in data:

print(\*i,sep=':')

else:

print('no record found')

**OUTPUT**

7369:SMITH:CLERK:7902:1980-12-17:800.00:None:20

7499:ALLEN:SALESMAN:7698:1981-02-20:1600.00:300.00:30

7521:WARD:SALESMAN:7698:1981-02-22:1250.00:500.00:30

7566:JONES:MANAGER:7839:1981-04-02:2975.00:None:20

7654:MARTIN:SALESMAN:7698:1981-09-28:1250.00:1400.00:30

7698:BLAKE:MANAGER:7839:1981-05-01:2850.00:None:30

7782:CLARK:MANAGER:7839:1981-06-09:2450.00:None:10

7788:SCOTT:ANALYST:7566:1982-12-09:3000.00:None:20

7839:KING:PRESIDENT:None:1981-11-17:5000.00:None:10

7844:TURNER:SALESMAN:7698:1981-09-08:1500.00:0.00:30

7876:ADAMS:CLERK:7788:1983-01-12:1100.00:None:20

7900:JAMES:CLERK:7698:1981-12-03:950.00:None:30

7902:FORD:ANALYST:7566:1981-12-03:3000.00:None:20

7934:MILLER:CLERK:7782:1982-01-23:1300.00:None:10

**9) Write a program to Record search using roll no**

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',passwd='tiger',database='class12’)

cur=mydb.cursor()

r=int(input('enter roll no'))

cur.execute('select\*from std where rollno={}'.format(r))

data=cur.fetchall()

print(data)

if data !=None:

print('record found - Details are. . . .')

for row in data:

print(\*row,sep=' : ')

ans=input('do u wish to update the class')

if ans=='y':

c=int(input('Enter new class'))

cur.execute('update std set class= {} where rollno = {}'.format(c,r))

con.commit()

print('$$ Record updated##')

else:

print('thank you')

else:

print('sorry no record found')

con.close()

**OUTPUT**

enter roll no7902

record found - Details are. . . .

7902:FORD:ANALYST:7566:1981-12-03:3000.00:None:20

do u wish to update the classno

thank you

‘

1**0) Write a Program to display sum of salaries.**

import sys

import mysql.connector as ms

mydb=ms.connect(host='localhost',user='root',passwd='tiger',database='class12’)

cur=mydb.cursor()

cur.execute('select sum(sal) as "total" from emp where deptno=10')

data=cur.fetchall()

for row in data:

print(\*row,sep=' : ')

**OUTPUT**

8750.0

**11) Write a program to update table values**

import sys

import mysql.connector as ms

mydb=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=mydb.cursor()

print('welcome to std database')

ans='y'

while ans=='y':

r=int(input('enter roll no'))

n=input('Enter name')

c=int(input('Enter class'))

s=input('enter sex')

query="insert into std values({0},{1},{2},{3})".format(r,n,c,s)

cur.execute(query)

con.commit()

print('$$ Record Saved ##')

ans=input('do you want to add another record')

**OUTPUT**

enter roll no2

Enter name Abraham

Enter class 12

enter sex m

$$ Record Saved ##

do you want to add another record no

**12) Write a program to record search using employment number**

import sys

import mysql.connector as ms

mydb=ms.connect(host='localhost',user='root',passwd='tiger',database='class12')

cur=mydb.cursor()

n=int(input('enter empno to search'))

cur.execute('select \* from emp where empno='+str(n))

data=cur.fetchone()

if data!=None:

print(\*data)

else:

print('No record found')

**OUTPUT**

enter empno to search7902

7902 Ford ANALYST 7566 1981-12-03 3000.00 None 20

**13) Write a Program to fetch 3 records**

import sys

import mysql.connector as ms

mydb=mysql.connector.connect(host='localhost',user='root',passwd='tiger',database='class12 ')

cur=mydb.cursor()

cur.execute('select \* from emp')

data=cur.fetchmany(3)

rec=cur.rowcount

print('Total records are',rec)

for row in data:

print(\*row,sep=' : ')

**OUTPUT**

Total records are 3

7369 : Smith : CLERK : 7902 : 1980-12-17 : 800.00 : None : 20

7499 : Allen : SALESMAN : 7698 : 1981-02-20 : 1600.00 : 300.00 : 30

7521 : Ward : SALESMAN : 7698 : 1981-02-22 : 1250.00 : 500.00 : 30

**14) Write a Program using fetchone**

import sys

import mysql.connector as ms

mydb=mysql.connector.connect(host='localhost',user='root',passwd='tiger',database='class12 ')

cur=mydb.cursor()

cur.execute('select \* from emp')

data=cur.fetchone()

for row in data:

print(row,sep=',',end=' ')

con.close()

**OUTPUT**

7369 Smith CLERK 7902 1980-12-17 800.00 None 20

**15) Write a Program using fetchall**

import sys

import mysql.connector as ms

mydb=mysql.connector.connect(host='localhost',user='root',passwd='tiger',database='class12 ')

cur=mydb.cursor()

cur.execute('select \* from emp')

data=cur.fetchall()

rec=cur.rowcount

print('the total records are:',rec)

for row in data:

print(\*row)

print()

**OUTPUT**

the total records are: 13

7369 SMITH CLERK 7902 1980-12-17 800.00 None 20

7499 ALLEN SALESMAN 7698 1981-02-20 1600.00 300.00 30

7521 WARD SALESMAN 7698 1981-02-22 1250.00 500.00 30

7566 JONES MANAGER 7839 1981-04-02 2975.00 None 20

7654 MARTIN SALESMAN 7698 1981-09-28 1250.00 1400.00 30

7698 BLAKE MANAGER 7839 1981-05-01 2850.00 None 30

7782 CLARK MANAGER 7839 1981-06-09 2450.00 None 10

7788 SCOTT ANALYST 7566 1982-12-09 3000.00 None 20

7839 KING PRESIDENT None 1981-11-17 5000.00 None 10

7844 TURNER SALESMAN 7698 1981-09-08 1500.00 0.00 30

7876 ADAMS CLERK 7788 1983-01-12 1100.00 None 20

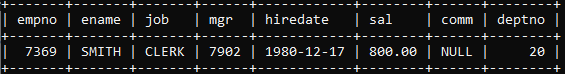
7900 JAMES CLERK 7698 1981-12-03 950.00 None 30

7934 MILLER CLERK 7782 1982-01-23 1300.00 None 1

**SQL QUERIES**

**1) WAQ to list the details about SMITH?**

**Ans:** Select \* from emp where ename=”smith”;

**Output: **

**2) WAQ to display employees whose names start with S.**

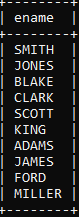
**Ans:** Select ename from emp where ename like “s%”;

**Output:**

****

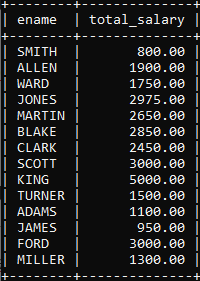
**3) WAQ to list out the employees who are not receiving any commission.**

**Ans:** Select ename from emp where comm is NULL;

**Output: **

**4) WAQ to display employee number and total salary for each employee.**

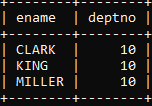
**Ans:** Select ename, (sal + ifnull(comm, 0)) as total\_salary from emp;

**Output: **

**5) WAQ to display names of all employees who are working in department no 10.**

**Ans:** select ename,deptno from emp where deptno="10";

**Output:**

****

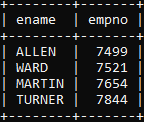
**6) WAQ to display names of all employees working as clerk and salary greater than 3000;**

**Ans:** select ename,job, sal from emp where job="clerk" and sal>3000;

**Output:** Empty set

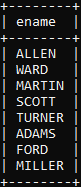
**7) WAQ to display employee numbers and names for employees who earn commission**

**Ans:** select ename,empno from emp where comm not like "Null";

**Output: **

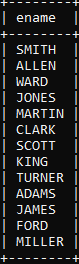
**8) WAQ to display the names of employees working as clerk,salesman or analyst and drawing salary more than 1000**

**Ans:** select ename from emp where job in ('CLERK','SALESMAN','ANALYST') and sal > 1000;

**Output: :**

**9) WAQ Display the names of employees working in the department number 10,20,40 or employees working as clerks, salesman or analyst.**

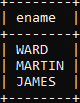
**Ans:** select ename from emp where deptno in(10,20,40) or job in ('clerk','salesman','analyst');

**Output: **

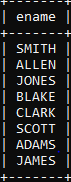
**10) WAQ to display the names of employees whose second letter is A**

**Ans:** select ename from emp where ename like '\_A%';

**Output:**

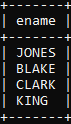


**11) WAQ to display the names of employees whose name is exactly five characters in Length**

**Ans:**

**12) WAQ to display the names of employees who are not working as salesman,clerk or analyst.**

**Ans:** select ename from emp where job not in ('SALESMAN', 'CLERK', 'ANALYST');

**Output:**

**13) WAQ to display the maximum salary being paid to clerk;**

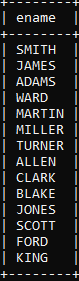
**Ans:** select max(sal) from emp where job like 'CLERK' ;

**Output:**

**14) WAQ to display the names of employees in order of their salaries**

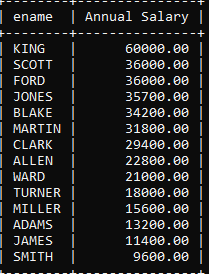
**Ans:** select ename from emp order by sal ;

**Output:**



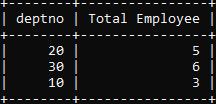
**15) WAQ to display the name of employees along with their annual salaries ,name of employee earning highest salary should come first**

**Ans:** select ename, (sal+ifnull(comm,0))\*12 as 'Annual Salary' from emp order by (sal+ifnull(comm,0))\*12 desc;

**Output:**

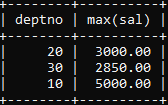
**16) WAQ to display dept numbers and total employees within each group**

**Ans:** select deptno, count(\*) as 'Total Employee' from emp group by deptno;

**Output:**

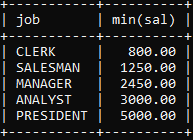
**17) WAQ to deptno and maximum salary for each department**

**Ans:** select deptno, max(sal) from emp group by deptno;

**Output:** 

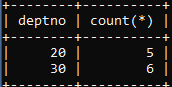
**18) WAQ to display each job along with minimum sal paid for that job**

**Ans:** select job, min(sal) from emp group by job;

**Output:**

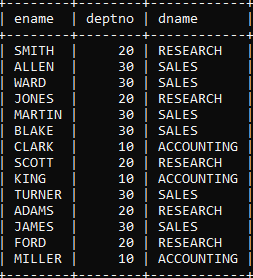
**19) WAQ to deptnos with more than three employees in each department**

**Ans:** select deptno, count() from emp group by deptno having count() >3;

**Output:**

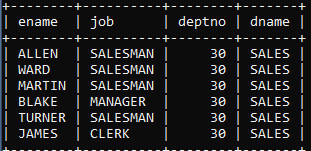
**20) WAQ to display name, deptno and dept name for all employees**

**Ans:**Select ename,emp.deptno,dname from emp,dept where emp.deptno=dept.deptno;

**Output: **

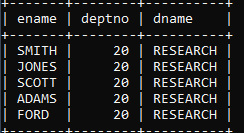
**21) WAQ to select ename,job,deptno and dname is sales**

**Ans:** select ename,job,dept.deptno,dname from emp, dept where emp.deptno = dept.deptno and dname='SALES';

**Output: **

**22) WAQ that will display the employee name,deptno and all the employees who work in the same department as smith**

**Ans:** Select ename,emp.deptno,dname from emp,dept where emp.deptno=dept.deptno and emp.deptno=(Select deptno from emp where ename=”smith”);

**Output: **

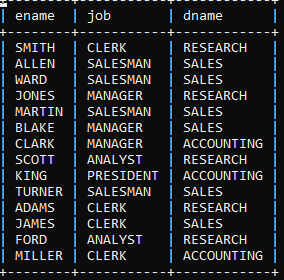
**23) WAQ to list the employees with their departments**

**Ans:**select ename, dname from emp, dept where emp.deptno = dept.deptno;

**Output: **

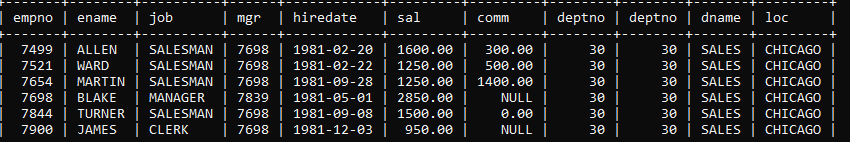
**24) WAQ to display employees with their designations with their jobs**

**Ans:** select ename,job,dname from emp,dept where emp.deptno=dept.deptno;

**Output:**

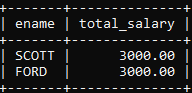
**25) WAQ to display employee details of sales and operations department**

**Ans:** select \* from emp, dept where dname in ('SALES', 'OPERATIONS') and emp.deptno = dept.deptno;

**Output: **

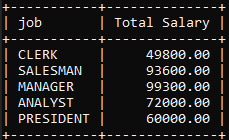
**26) Display the total salary of analyst from dept 20**

**Ans:** Select ename, (sal + ifnull(comm, 0)) as total\_salary from emp where deptno=”20” and job=”analyst”;

**Output:** 

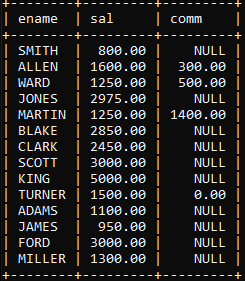
**27) Dispay the various job along with total salary for each job where total sal >40000**

**Ans:** select job, sum((sal+ifnull(comm,0))\*12) as 'Total Salary' from emp group by job having sum((sal+ifnull(comm,0))\*12) > 40000;

**Output:** 

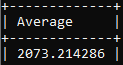
**28) Write a query to list out name, salary, commission for all employees:**

**Ans:** select ename, sal, comm from emp;

**Output:** 

**29) WAQ to calculate average salaries of all employees**

**Ans:** select avg(sal) as Average from emp;

**Output: **

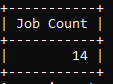
**30) WAQ to count total number of records**

**Ans:** select count(\*) as Total from emp;

**Output:** 

**31) WAQ to find the number of jobs**

**Ans :** select count(job) as “ Job Count” from emp;

**Output:** 

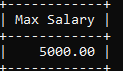
**32) WAQ to find the number of distinct jobs**

**Ans:** select count(distinct job) as “Distinct Jobs” from emp;

**Output: Picture 118**

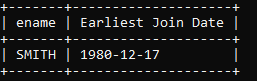
**33) WAQ to display the maximum salary**

**Ans:** select max(sal) as “Max Salary” from emp;

**Output:** 

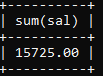
**34) WAQ to display to joining date of the senior most employee**

**Ans :** select ename, min(hiredate) as “Earliest Join Date” from emp;

**Output: **

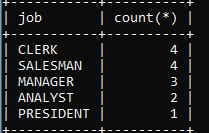
**35) WAQ to display total salary of all employees who have 5 letters in their name**

**Ans:** select sum(sal) from emp where ename like “\_\_\_\_\_”;

**Output:** 

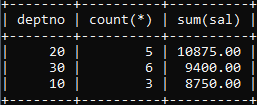
**36) WAQ to calculate no of employees in each job;**

**Ans: Select job,count(\*) from emp group by job;**

**Output: **

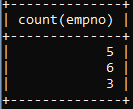
**37) WAQ to display no of employees in each department with their salary**

**Ans:** Select deptno,count(\*),sum(sal) from emp group by deptno;

**Output: **

**38) WAQ to count the number of employees in each group of department**

**Ans:** Select count(empno) from emp group by deptno;

**Output: **

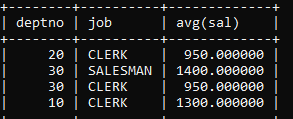
**39) WAQ to display dept no and average salary for employees having avg commission more than 100 and average salary more than 1800**

**Ans:** Select deptno,avg(comm),avg(sal) from emp group by deptno having avg(comm)>100 and avg(sal)>1800;

**Output: Empty set**

**40) WAQ to display deptno , job and average salary of salesmen and clerks**

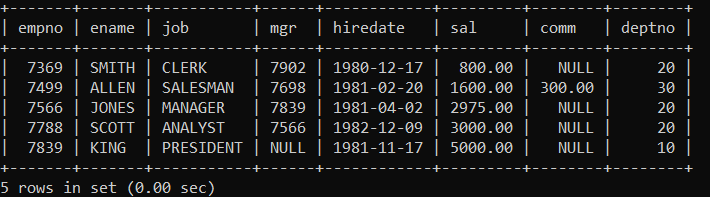
**Ans:**Select deptno,job,avg(sal) from emp group by deptno,job having job in (“clerk”,”salesman”);

**Output: **

**41) Write a query to display the table jobwise within a department**

**Ans:** Select \* from emp group by job;

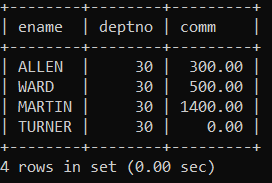
**Output:**

****

**42) Write a query to display number and names of employees with commission**

**Ans:** select ename,deptno,comm from emp where comm is NOT NULL;

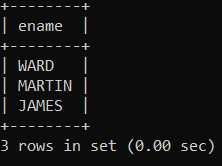
**Output:**

****

**43) Write a query to display names of employees whose names have second alphabet**

**Ans:** Select ename from emp where ename like '\_a%' ;

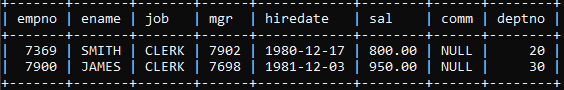
**Output:**

****

**44) Write a query to list the employees whose salary is of length 6 and ending with 0**

**Ans:** select \* from emp where length(sal)=6 and sal like "%0";

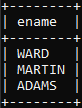
**Output:**

****

**45) Write a query to list the employees whose salary is ranging from 12000 and 15000**

**Ans:** select ename from emp where sal\*12 between 12000 and 15000;

**Output:**

****

**46) Write a query to list the employees whose names have the character “LL” together**

**Ans:** select ename from emp where ename like "%LL%";

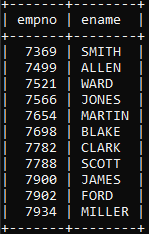
**Output:**

****

**47) Write a query to list the employees whose empno is not starting with the digit 78**

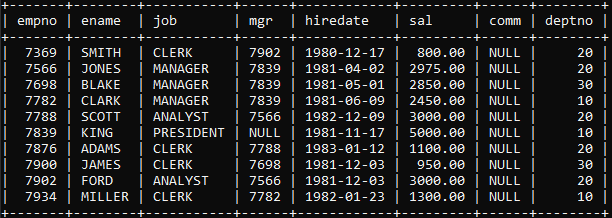
**Ans:** Select empno, ename from emp where empno not like"78%";

**Output:**

****

**48) Write a query to list the employees whose jobs are same as “miller” or salary is more than “allen”**

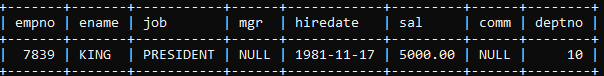
**Ans:** select \* from emp where job=(select job from emp where ename="miller") or sal>(select sal from emp where ename="allen");

**Output: **

**49) Write a query to find the details of the highest paid employee**

**Ans: s**elect \* from emp where sal in (select max(sal) from emp);

**Output:**

****

**50) Write a query to list the employees whose salary is equal to the average of max and minimum**

**Ans:** select \* from emp where sal=(select(max(sal)+ min(sal))/2 from emp);

**Output:**

****