

**KENDRIYA VIDYALAYA SANGATHAN**

**A**

**Practical File of**

**Computer Science (083)**

**Submitted for**

**Class-XII**

**at**

# KENDRIYA VIDYALAYA, COOCHBEHAR

**(COB)**

**For the Session (2022-2023)**

|  |  |
| --- | --- |
| **Submitted to** | **Submitted by** |
| Mr Govind Prasad Arya | Snehasish Mandal |
| PGT-Computer Science | Class-XII - B |
| Kendriya Vidyalaya | Roll Number : 30 |
| CoochBehar (COB), WB | KV- COB, WB |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LABORATORY: COMPUTER SCIENCE (083), CLASS: XII** | | | | |
| **Index Page** | | | | |
| **SN** | **Date** | **Objective** | **Page**  **No** | **Teacher’s Signature** |
| **FUNCTION BASED PROGRAMMING** | | | | |
| **1** |  | WAP to build a menu-based calculator for basic mathematical operations using functions. |  |  |
| **2** |  | WAP to calculate the factorial of an entered number using function. |  |  |
| **3** |  | WAP to print Fibonacci series till N terms. Number N will be entered by the user. |  |  |
| **4** |  | WAP to calculate factorial of an entered number using recursion. |  |  |
| **5** |  | WAP to calculate the value of the expression.  y=six(x) + cos(x) where x will be entered by the user. |  |  |
| **6** |  | WAP to calculate the value of the following Taylor series-  Y=x1/1! + x2/2! + x3/3! +x4/4! + x5/5! + ……+n Where, n will be entered by the user. |  |  |
| **7** |  | WAP to store any N numbers into a list using append method and print the sum of available numbers in the list. |  |  |
| **8** |  | WAP to store any N numbers into a list using append and pass that list to a function and print the greatest and smallest number available in the list. |  |  |
| **9** |  | WAP to print Fibonacci series till N terms using recursion. Number N will be entered by the user. |  |  |
| **10** |  | WAP to enter N elements into a list then search an entered element in the list using linear search method. |  |  |
| **11** |  | WAP to enter N elements into a list then search an entered element in the list using binary search method. |  |  |
| **12** |  | WAP to show the effects of all the functions available in math library |  |  |
| **13** |  | WAP to show the effects of all string’s functions. |  |  |
| **FILE BASED PROGRAMMING** | | | | |
| **14** |  | Write a python program to read a text file  “INPUT.TXT” and print its entire content as output in console. Also print total number of characters, words and lines available in the file. |  |  |
| **15** |  | Write a python program to read a text file  “INPUT.TXT” and write its content into another file “OUTPUT.TXT”. Skip the space and new line character while writing to the output file. |  |  |
| **16** |  | Write a python program to read a text file |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | “INPUT.TXT” and write it’s character’s summary (number of lower case, upper case, new lines, white spaces, digits, special characters and total number of characters) into another file “OUTPUT.TXT”. |  |  | |
| **17** | |  | Write a python program to read information of any N students (name, roll number and age) from the keyboard and write these information in a text file “OUTPUT.TXT” in coma separated values (CSV) format. |  |  | |
| **18** | |  | Read a text file “INPUT.TXT” which contains roll number, name and age of students. Each record has written in a separate line. Write a python program to find an entered roll number and update the name of that student with a new entered name. Display a message “The roll number is not available” in case the file does not contains that roll number. |  |  | |
|  | | **DATA STRUCTURE’S PROGRAMMING** | | | | |
| **19** | |  | WAP to implement PUSH, POP and DISPLAY functions of a stack using list. |  |  | |
| **20** | |  | WAP to implement push function of a stack and then print all prime numbers available in the stack using display function. |  |  | |
| **21** | |  | WAP to implement INSERT, DELETE and  DISPLAY functions of a linear queue using list. |  |  | |
| **22** | |  | WAP to implement INSERT function of a linear queue and then print available numbers which are divisible by 3. |  |  | |
| **23** | |  | WAP to insert names of any N candidates in a queue in a bank and then search an entered name. Display the result in the form of – NAME found or not found. |  |  | |
| **24** | |  | Write python program to read data from a CSV file and write its content in a table using SQL queries. |  |  | |
|  |  | **DATABASE PROGRAMMING** | | | |  |
| **25** | |  | Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table. Print all the details available in the table. |  |  | |
| **26** | |  | Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table. Delete the details of a particular student from the table. |  |  | |
| **27** | |  | Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table.  Use the select command to get the details of the students with marks between 60 to 80. |  |  | |
| **28** | |  | Create a student table with the student id, name, and |  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | marks as attributes where the student id is the primary key. Insert any five records in the table. Write SQL query to modify the details of a particular student in above table. |  |  |
| **29** |  | A student table contains roll number, name and age of all the students of a class. Write python program (using SQL query) to search an entered name into the table. |  |  |
| **30** |  | Create a new table (name, date of birth) by joining two tables (student id, name) and (student id, date of birth).  Display all the details available in new table. |  |  |
| **31** |  | Create a new table (order ID, customer Name, and order Date) by joining two tables (order ID, customer ID, and order Date) and (customer ID, customer Name, contact Name, country). |  |  |
| **32** |  | A student table contains roll number, name and age of all the students of a class. Write python program (using SQL aggregate functions) to print min, max, sum and average age of the student in the table. |  |  |
| **33** |  | An employee table contains Employee\_ID, name, age and salary of employees of a company. Write python program (using SQL query) to for the following-   1. Display name in descending order whose age is more than 40. 2. Display the average salary grouped by age. 3. Display the details of employee where salary is between 50000 and 80000. 4. Display the sum of salaries of employees whose name contains “sha”. 5. Display all the details of employees group by age having count(age)>3 |  |  |

**FUNCTION BASED PROGRAMMING**

**PRACTICAL:-01 Date:-**

**Objective:**

**WAP to build a menu-based calculator for basic mathematical operations using functions.**

**ALGORITHM:**

1. Open python idle.

1. Make different functions to make calculations easily.Functions like add,multiply,subtraction,division,

1. Using global fuctions input two numbers for calculations.

1. Make definition for add functions.

**CODE:**

def add(): print(n1+n2)

def sub(): print(n1-n2)

def multiply(): print(n1\*n2)

def division(): print(n1/n2)

def inputt(): global n1 , n2

n1 = int(input('Enter 1st no. -'))

n2 = int(input('Enter 2nd no. -'))

print('For,\nAddition - a\nSubtraction  
 -b\nDivision - c\nMultiplication - d\nExit x')

while (1): inpt = input('Your choice - ')

if inpt == 'a': inputt() , add()

if inpt == 'b': inputt() , sub()

if inpt == 'c': inputt() , division()

if inpt == 'd': inputt() , multiply()

if inpt == 'x':

break

**OUTPUT:**

For,

Addition - a

Subtraction - b

Division - c

Multiplication - d

Exit - x

Your choice - a

Enter 1st no. -1

Enter 2nd no. -9

10

Your choice - d

Enter 1st no. -2

Enter 2nd no. -8

16

Your choice - x

**PRACTICAL:-02 Date:-**

**Objective:**

**WAP to calculate the factorial of an entered number using function.**

**ALGORITHM:**

1. Make a function named factorial or something similar with a number as a parameter .
2. Return (factorial(no)).
3. Now, take a number from the user as input and store it in a variable (num).
4. Import math module.
5. Call the function with num and print it .

**CODE:**from math import \*

num = int(input('Enter a number - '))   
def factoria(no):

return (factorial(no))

print(f 'factorial of (num) is (factorial(num))')

**OUTPUT:**

Enter a number - 4

factorial of 4 is 24

**PRACTICAL:-03 Date:-**

**Objective:**

**WAP to print Fibonacci series till N terms. Number N will be entered by the user.**

**ALGORITHM:**

1. Take a number from the user as input and store it in a variable (num).
2. Put a=0 and b=1 OR take any variable you want.
3. Use a loop and use range (num) .
4. Now, inside the loop take a variable (c) = a+b and print (c) .
5. Assign a=b and b=c.

**CODE:**

num = int(input('enter a number - '))

a = 0

b = 1

print(a,b,end = ' ') for i in range(num): c = a+b print(c, end=' ') a = b b = c

**OUTPUT:**

enter a number - 12

0 1 1 2 3 5 8 13 21 34 55 89 144 233

**PRACTICAL:-04 Date:-**

**Objective:**

**WAP to calculate factorial of an entered number using**

**recursion.**

**ALGORITHM:-**

**AlGO\_FACT(n)**

1. If n=1 then return 1, otherwise do step 2 to 3.
2. F=n \* ALGO\_FACT (n-1).
3. Return f .

**ALGO\_MAIN PROGRAM**

1. Input any number and store in n.
2. Call function FACT with value n and store return value into R.
3. Print value of R.

**CODE:-**

**>>**def fact(n):

**>>** if (n==1):

**>>** return(1)

**>>** else:

**>>** f=n\*fact(n-1)

**>>** return(f)

**>>**n=5

**>>**R=fact(n)

**>>**print(R)

**OUTPUT:-**

**>>**120

**PRACTICAL:-05 Date:-**

**Objective:**

**WAP to calculate the value of the expression.**

**y=six(x) + cos(x) where x will be entered by the user.**

**ALGORITHM:**

1. Import sin ,cos, radians from math .
2. Take num as an input and convert it into integer .
3. Put the num into sin and cos function and add them .
4. Print the result .

**CODE:**

import math

num = int(input('enter a number - ')) print ( math.sin(num) + math.cos(num))

**OUTPUT:**

enter a number – 0

1.0

**PRACTICAL:-06 Date:-**

**Objective:**

**WAP to calculate the value of the following Taylor series-**

**Y=x1/1! + x2/2! + x3/3! +x4/4! + x5/5! + ……+n Where, n will be entered by the user.**

**ALGORITHM:**

1. Import math .
2. Take num as an input and convert it into integer .
3. Use a loop and use range (1,num+1) .
4. Now, inside the loop take a variable (value) = value + ((num\*\*i)/(factorial(i))) .
5. Print the result .

**CODE:**

from math import \*

num = int(input('enter a number - '))

value = 0

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

value = value + ((num\*\*i)/(factorial(i)))

print (f' value = {value}')

**OUTPUT:**

enter a number - 2

value = 4.0

**PRACTICAL:-07 Date:-**

**Objective:**

**WAP to store any N numbers into a list using append method and print the sum of available numbers in the list.**

**ALGORITHM:**

1. Take num as an input and convert it into integer .
2. Create an empty list.
3. Use a loop and use range (num) .
4. Now, inside the loop take no as an input and convert it into integer .
5. Append no into the list.
6. Print the result .

**CODE:**

**>>**num = int(input('Enter the number of elements in the list - '))

**>>**list = []

**>>**for i in range (num):

**>>** no = int(input('enter number - '))

**>>** list.append(no)

**>>** print('your list' ,list)

**>>** print('Sum of the elements of the list is ', sum(list))

**OUTPUT:**

Enter the number of elements in the list - 4

enter number - 1

enter number - 2

enter number - 3

enter number - 4

your list [1, 2, 3, 4]

Sum of the elements of the list is 10

**PRACTICAL:-08 Date:-**

**Objective:**

**8. WAP to store any N numbers into a list using append and pass that list to a function and print the greatest and smallest number available in the list.**

**ALGORITHM:**

1. Take num as an input and convert it into integer .
2. Create an empty list.
3. Use a loop and use range (num) .
4. Now, inside the loop take no as an input and convert it into integer and append no into the list.
5. Append no into the list.
6. Print min(list) for minimum value and max(list) for maximum value .

**CODE:**

**>>**num = int(input('Enter the number of elements in the list - '))

**>>**list = []

**>>**for i in range (num):

**>>** no = int(input('enter number - '))

**>>** list.append(no)

**>>** print('your list is ' ,list)

**>>** print('minimum value is ', min(list),'\nmaximum value is ', max(list)) minmax()

**OUTPUT:**

Enter the number of elements in the list - 2

enter number - 1

enter number - 2

your list is [1, 2]

minimum value is 1 maximum value is 2

**PRACTICAL:-09 Date:-**

**Objective:**

**WAP to print Fibonacci series till N terms using recursion. Number N will be entered by the user.**

**ALGORITHM:**

1. Make fibo as function
2. Use if else statement
3. Take two cases as if n==0 and n==1 then it will return n
4. Else it use its fibonnaci calculating formula
5. Take a variable n to take input from user as to calculate the no of inputs
6. Put it in loop to calculate range
7. Call the fibo function
8. Print all the elements of Fibonacci series.

**CODE:**

**>>**def fibo(n):

**>>** if(n==0) or (n==1):

**>>** return n

**>>** else:

**>>** t=fibo(n-1) + fibo(n-1)

**>>** return t

**>>** n=int(input(“Enter no terms to calculate the fibonnaci series:”))

**>>** for i in range (n):

**>>** r=fibo(i)

**>>** print(r,end= “ ”)

**OUTPUT:**

Enter no terms to calculate the fibonnaci series : 4

0

1

1

2

**PRACTICAL:-10 Date:-**

**Objective:**

**WAP to enter N elements into a list then search an entered element in the list using linear search method.**

**ALGORITHM:**

1. To enter N elements into a list follow the steps of the previous program.
2. Take ln (the no. you wanted to find ) as an input and convert it into integer .
3. Use a loop and use range (num) for index .
4. Now, inside the loop take if (list[index] == ln ) then, print (f'{ln} found at index

{index}')

1. else print (f'{ln} not found').

**CODE:**

num = int(input('Enter the number of elements in the list - '))

list = []

for i in range (num):

no = int(input('enter number - '))

list.append(no)

ln = int(input('Enter the number you wanted to find - '))

for index in range (num):

if list[index] == ln :

print (f'{ln} found at index {index}')

Break

else :

print(f'{ln} not found')

**OUTPUT:**

Enter the number of elements in the list - 3

enter number - 1

enter number - 2

enter number - 3

Enter the number you wanted to find - 10

10 not found

**PRACTICAL:-11 Date:-**

**Objective:**

**WAP to calculate the factorial of an entered number using function.**

**ALGORITHM:**

1. Make a function named factorial or something similar with a number as a parameter .
2. Return (factorial(no)).
3. Now, take a number from the user as input and store it in a variable (num).

4.Import math module.

5. Call the function with num and print it .

**CODE:**

from math import \*

num = int(input('enter a number - '))

def factoria(no):

return (factorial(no))

print(f'factorial of {num} is {factoria(num)}')

**OUTPUT:**

enter a number - 4 factorial of 4 is 24

**PRACTICAL:-12 Date:-**

**Objective:**

**WAP to show the effects of all the functions available in math library .**

**CODE:**

r=5

import math

a=math.sqrt(r) # Square root calculation

print(a)

b=math.ceil(r) #Returns the smallest integer greater than or

print(b) equal to r.

c=math.fabs(r) #Returns the absolute value of r

print(c)

d=math.factorial(r) #Returns the factorial of r

print(d)

e=math.floor(r) #Returns the largest integer less than or equal to r

print(e)

f=math.isfinite(r) #Returns True if r is neither an infinity nor a NaN print(f) (Not a Number)

g=math.pow(r, 4) #Returns r raised to the power 4

print(g)

**OUTPUT:**

2.23606797749979

5

5.0

120

5

625.0

**PRACTICAL:-13 Date:-**

**Objective:**

**WAP to show the effects of all string’s functions.**

**CODE:**

s= str(“My Name is Snehasish Mandal student of computer science”)

a=str.capitalize(s)

print(a)

b= str.upper(s) #Converts all lowercase characters in a string into uppercase

print(b)

c= str.title(s) #Convert string to title case

print(c)

d= str.splitlines(s) #Split the lines at line boundaries

print(d)

e= str.lower(s) #Converts all uppercase characters in a string into lowercase

print(e)

f= str. isdecimal(s) #Returns true if all characters in a string are decimal

print(f)

g= str.isdigit(s) #Returns “True” if all characters in the string are digits

print(g)

h= str.isidentifier(s)#Check whether a string is a valid identifier or not

print(h)

i= str.islower(s) #Checks if all characters in the string are lowercase

print(i)

j= str.isnumeric(s) #Returns “True” if all characters in the string are numeric print(j) characters

k= str.isprintable(s)#Returns “True” if all characters in the string are printable or print(k) the string is empty

l= str.isspace(s) #Returns “True” if all characters in the string are whitespace print(l) characters

m= str.istitle(s) #Returns “True” if the string is a title cased string

print(m)

n= str.isupper(s) #Checks if all characters in the string are uppercase

print(n)

**OUTPUT:**

My name is snehasish mandal student of computer science

MY NAME IS SNEHASISH MANDAL STUDENT OF COMPUTER SCIENCE

My Name Is Snehasish Mandal Student Of Computer Science

['My Name is Snehasish Mandal student of computer science']

my name is snehasish mandal student of computer science

False

False

False

False

False

True

False

False

False

**FILE BASED PROGRAMMING**

|  |
| --- |
|  |
|  |  |

**PRACTICAL:-14 Date:-**

**Objective:**

**Write a python program to read a text file “INPUT.TXT” and print its entire content as output in console. Also print total number of characters, words and lines available in the file.**

**ALGORITHM:**

1. open “INPUT.txt” file with read mode
2. read the data and store in a variable (cons)
3. Print the following print('File content=\n',cons)
4. print('Total number of characters = ',len(cons))
5. print('Total number of newline characters =',len([i or i in cons if i=='\n']))
6. print(len([i for i in cons.split() if i.isalpha()]))

**CODE:**

file = open('INPUT.txt','r')

cons = file.read()

print('File content =\n',cons)

print('Total number of characters = ',len(cons))

print('Total number of newline characters =',len([i for i in cons if i== '\n'])) print(len([i for i in cons.split() if i.isalpha()]))

**OUTPUT:**

File content =

Gnocchi are a varied family of dumplings in Italian cuisine . They are made of small lumps of dough composed of semolina , ordinary wheat,flour , egg .

Total number of characters = 150

Total number of newline characters = 2

Total number of words = 25

**PRACTICAL:-15 Date:-**

**Objective:**

**Write a python program to read a text file “INPUT.TXT” and write its content into another file “OUTPUT.TXT”. Skip the space and new line character while writing to the output file.**

**ALGORITHM:**

1. Open the “INPUT.txt” file with read mode .
2. read the data and store it in a variable (cons).
3. Now open the “OUTPUT.txt” file with write mode .
4. for chr in cons.
5. if chr != ' ' and chr != '\n' then write(chr) in file OUTPUT.txt.

**CODE:**

file = open('INPUT.txt','r')

cons = file.read()

file2 = open('OUTPUT.txt','w')

for chr in cons: if chr != ' ' and chr != '\n': file2.write(chr) file2.close()

**File INPUT.txt -(input file)**

In friction when two bodies are rubbed together,electrons are transferred from one body to the other.This makes one body become positively charged whilethe other become negatively charged.

**File OUTPUT.txt - (output file)**

Infrictionwhentwobodiesarerubbedtogether,electronsaretransferredfromonebodytotheother.Thismakesonebodybecomepositivelychargedwhiletheotherbecomenegativelycharged.

**PRACTICAL:-16 Date:-**

**Objective:**

**“INPUT.TXT” and write it’s character’s summary (number of lower case, upper case, new lines, white spaces, digits, special characters and total number of characters) into another file “OUTPUT.TXT”.**

**ALGORITHM:**

1. initializes lowercase, uppercase, newlines, spaces, digits, special characters to zero OR you can take any variable you want.
2. open the “INPUT.txt” in read mode and read the file and store the data in variable (chr)
3. loop through the data and check if the element is one of the given initials and if it is the increment of that variable with 1.
4. open “output.txt” in write mode and write the summary with appropriate text.

**CODE:**

a=b=c=d=e=f=g =0

file = open ('INPUT.txt','r')

file2 = open ('OUTPUT.txt','w')

chr = file.read()

for i in chr :

f = f+1

if i.isdigit() : a = a+1

elif i.islower(): b = b+1

elif i.isupper(): c = c+1

elif i == '\n': d = d+1

elif i == ' '

e = e+1

file2.write(f'Total no. of characters :{f}\n')

file2.write(f'Total no. of lowercase chr :{b}\n')

file2.write(f'Total no. of uppercase chr :{c}\n')

file2.write(f'Total no. of newline chr :{d}\n')

file2.write(f'Total no. of spaces :{e}\n')

file2.write(f'Total no. of digits :{a}')

file2.close()

**OUTPUT:**

Total no. of character :149

Total no. of lowercase chr :114

Total no. of uppercase chr :4

Total no. of newline chr :2

Total no. of spaces :25

Total no. of digits :0

**PRACTICAL:-17 Date:-**

**Objective:**

**Write a python program to read information of any N students (name, roll number and age) from the keyboard and write these information in a text file “OUTPUT.TXT” in coma separated values (CSV) format.**

**CODE:**

import csv

fw=open("D:\\Holidays Homework\\snehasish.csv",'w',newline='')

cw=csv.writer(fw)

cw.writerow(['NAME','ROLL','AGE'])

N=int(input('Enter no of times you want to write; '))

for i in range(N):

nm=input('Enter Name: ')

rl=int(input('Enter Roll No : '))

age=int(input('Enter Age: '))

cw.writerow([nm,rl,age])

fw.close()

**OUTPUT:**

Enter no of times you want to write: 2

Enter Name: Snehasish

Enter Roll No : 30

Enter Age: 15

Enter Name: Ayshi

Enter Roll No : 13

Enter Age: 18

**PRACTICAL:-18 Date:-**

**Objective:**

**Read a text file “INPUT.TXT” which contains roll number, name and age of students. Each record has written in a separate line. Write a python program to find an entered roll number and update the name of that student with a new entered name. Display a message “The roll number is not available” in case the file does not contains that roll number.**

**CODE:**

fr=open("D:\\Holidays Homework\\snehasish.csv",'r')

fw=open("D:\\Holidays Homework\\snehasish.csv",'w')

L=fr.readlines()

rn=int(input('Enter Roll No to Search: '))

f=0

for line in L:

e=line.split('\t')

if e[1]==str(rn):

nm=input('Enter Name')

e[2]=nm+'\n'

f=1

fw.write(e[0])

fw.write('\t')

fw.write('\t')

fw.write(e[2])

if f==0:

print("The Roll no is not available")

fw.close()

**OUTPUT:**

Enter Roll No to Search:

**DATA STRUCTURE’S PROGRAMMING**

**PRACTICAL:-19 Date:-**

**Objective:**

**WAP to implement PUSH, POP and DISPLAY functions of a stack using list.**

**ALGORITHM:**

1. Make three functions as PUSH,POP and DISPLAY.
2. First make the PUSH function using python program.
3. Then make the POP function using python program.
4. Then do Menu Programming.
5. Using the following functions use the menu programming.
6. Break the program.
7. Print and Exit it from the programs.

**CODE:**

def push (element):

if(len(L)==Max):

print("Stack overflow")

else:

L.append(element)

def pop():

if(len(L)==0):

print("Stack Underflow")

else:

e=L.pop()

print("Top element %d has popped",e)

def display():

if(len(L)==0):

print("No element of Display")

else:

for e in L :

print(e)

#MENU BASED PROGRAMING

L=[]

Max=5

while(True):

print("1:PUSH \n 2:POP \n 3:DISPLAY \n 4:EXIT ")

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

if (choice==1):

e=int(input("Enter Element: "))

push(e)

elif (choice==2):

pop()

elif (choice==3):

display()

elif (choice==4):

break

**OUTPUT:**

1:PUSH

2:POP

3:DISPLAY

4:EXIT

Enter your choice:

**(Select the following Programs as your wish)**

**PRACTICAL:-20 Date:-**

**Objective:**

**WAP to implement push function of a stack and then print all prime numbers available in the stack using display function.**

**CODE:**

def push (element):

if(len(L)==Max):

print("Stack overflow")

else:

L.append(element)

def display():

if(len(L)==0):

print("Stack Underflow")

else:

for e in L :

if e>1 :

for i in range (2,int(x/2)+1):

if x%i==0:

break

else:

print(x,end='')

else:

continue

#MENU BASED PROGRAMING

L=[]

Max=10

while(True):

print("1:PUSH \n 2:TO PRINT THE PRIME NUMBERS \n 3:TO EXIT")

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

if (choice==1):

e=int(input("Enter Element: "))

push(e)

elif (choice==2):

display()

elif (choice==3):

break

**OUTPUT:**

1:PUSH

2:TO PRINT THE PRIME NUMBERS

3:TO EXIT

Enter your choice:

**PRACTICAL:-21 Date:-**

**Objective:**

**WAP to implement INSERT, DELETE and DISPLAY functions of a linear queue using list.**

**ALGORITHM:**

1. Make three functions as INSERT(ENQUEUE),DELETE(DEQUEUE) and DISPLAY.
2. First make the INSERT(ENQUEUE) function using python program.
3. Then make the DELETE(DEQUEUE) function using python program.
4. Then do Menu Programming.
5. Using the following functions use the menu programming.
6. Break the program.
7. Print and Exit it from the programs.

**CODE:**

def ENQUEUE (element):

if(len(L)==Max):

print("Queue overflow")

else:

L.append(element)

def DEQUEUE():

if(len(L)==0):

print("Queue Underflow")

else:

e=L.pop()

print("Top element %d has DELETED",e)

def DISPLAY():

if(len(L)==0):

print("No element to DISPLAY")

else:

for e in L :

print(e)

L=[]

Max=5

while(True):

print(" 1:ENQUEUE \n 2:DEQUEUE \n 3:DISPLAY \n 4:EXIT ")

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

if (choice==1):

e=int(input("Enter Element: "))

ENQUEUE(e)

elif (choice==2):

DEQUEUE()

elif (choice==3):

DISPLAY()

elif (choice==4):

break

**OUTPUT:**

1:ENQUEUE

2:DEQUEUE

3:DISPLAY

4:EXIT

Enter your choice:

**(Select the following Outputs as your wish)**

**PRACTICAL:-22 Date:-**

**Objective:**

**WAP to implement INSERT function of a linear queue and then print available numbers which are divisible by 3.**

**CODE:**

def insert():

a=int(input('Enter the element of the list: '))

L.append(a)

def search():

for i in L:

if(i%3==0):

C.append(i)

print(C)

L=[]

C=[]

for i in range(1,5):

insert()

search()

**OUTPUT:**

Enter the element of the list: 3

Enter the element of the list: 6

Enter the element of the list: 10

Enter the element of the list: 15

[3, 6, 15]

**PRACTICAL:-23 Date:-**

**Objective:**

**WAP to insert names of any N candidates in a queue in a bank and then search an entered name. Display the result in the form of – NAME found or not found.**

**CODE:**

def ENQUEUE (element):

L.append(element)

def DISPLAY(0):

f=0

for i in L:

if i==0:

print('Name Found')

f=1

elif f==0:

print('Name not Found')

L=[]

t=int(input('Enter the No of Candidate: '))

for x in range(t):

nm=input('Enter the Name of the Candidate: ')

ENQUEUE(nm)

y=input('Enter the name to be Searched: ')

DISPLAY(y)

**OUTPUT:**

Enter the No of Candidate: 2

Enter the Name of the Candidate: Mr Snehasish Mandal

Enter the Name of the Candidate: Miss Ayshi Poddar

Enter the name to be Searched: Mr Snehasish Mandal

Name Found

**DATABASE PROGRAMMING**

**PRACTICAL:-25 Date:-**

**Objective:**

**Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table.Print all the details available in the table.**

**ALGORITHM :**

1. Create a database if you don't have by “CREATE DATABASE ”
2. Then use that database by “USE ”
3. Then to create a table by “CREATE TABLE < name of table> (ID int NOT NULL

Name varchar() NOT NULL, marks int, PRIMARY KEY (ID) )

1. Now to add data “INSERT INTO (id, Name, marks) VALUES (<id>,<name>,<marks> )” and to add multiple values separate each by comma.

5. Use (SELECT \* FROM database\_name.Table\_name) to print the content of the table.

**CODE :**

**To create table-**

USE database;

CREATE TABLE `database`.`student table` (

`student\_id` INT NOT NULL,

`Name` VARCHAR(30) NOT NULL,

`Marks` INT NOT NULL, PRIMARY KEY (`student\_id`));

**To insert data-**

USE database;

INSERT INTO `database`.`student table` (`student\_id`, `Name`, `Marks`)

VALUES

('1 ', 'Manish ' , '600'),

('2' , 'Pandit ' , '599'),

('3 ', 'Subhro' , '598'),

('4' , 'Ayshi' , '600'),

('5' , 'Killua' , '599');

**To retrieve data -**

USE database;

SELECT \* FROM database.`student table`;

**OUTPUT:**

+------------+--------+------------+

| student\_id | Name | Marks |

+------------+--------+------------+

| 1 | Manish | 600 |

| 2 | Pandit | 599 |

| 3 | Subhro | 598 |

| 4 | Ayshi | 600 |

| 5 | Killua | 599 |

+------------+--------+------------+

**PRACTICAL:-26 Date:-**

# Refer Practical no 25 for table and Database :-

**Objective:**

# Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table. Delete the details of a particular student from the table.

**ALGORITHM:**

1. Use that database by “USE ”
2. Then delete a record by “DELETE FROM < name of table> WHERE (‘id’ = 'id of the student whose data you wanted to delete')
3. Use (SELECT \* FROM database\_name.Table\_name) to print the content of the table.

**CODE :**

USE database;

DELETE FROM `database`.`student table` WHERE (`student\_id` = '5');

SELECT \* FROM database.`student table`;

**OUTPUT:**

+------------+--------+------------+

| student\_id | Name | Marks |

+------------+--------+------------+

| 1 | Manish | 600 |

| 2 | Pandit | 599 |

| 3 | Subhro | 598 |

| 4 | Ayshi | 600 |

+------------+--------+------------+

**PRACTICAL:-27 Date:-**

**Refer Practical no 25 for table and Database :-**

**Objective:**

**Create a student table with the student id, name, and marks as attributes where the student id is the primary key.**

**Insert any five records in the table.**

**Use the select command to get the details of the students with marks between 60 to 80.**

**ALGORITHM:**

1. Use that database by “USE ”
2. Then get data from a particular range by “SELECT \* FROM database.`name of the table` WHERE 60 < Marks < 80;”

**CODE:**

Use database;

SELECT \* FROM database.`student table` WHERE 60 < Marks < 80;

**OUTPUT:**

+------------+--------+----------+

| student\_id | Name | Marks |

+------------+--------+----------+

| 1 | Nobara | 79 |

| 4 |kurapika| 75 |

+------------+--------+----------+

**PRACTICAL:-28 Date:-**

# Refer Practical no 25 for table and Database :-

**Objective:**

**Create a student table with the student id, name, and marks as attributes where the student id is the primary key. Insert any five records in the table. Write SQL query to modify the details of a particular student in above table.**

**ALGORITHM:**

1. Use that database by “USE ”
2. Use ‘UPDATE `name of the table` SET Name = 'updated\_name', `Marks` = 'updated\_marks' WHERE (`student\_id` = 'id of the student whose details you wanted to change') ;

**CODE:**

USE database;

UPDATE `student table` SET Name = 'Armin', `Marks` = '597' WHERE

(`student\_id` = '3') ;

SELECT \* FROM `student table;

**OUTPUT:**

+------------+--------+------------+

| student\_id | Name | Marks |

+------------+--------+------------+

| 1 | Sunami | 600 |

| 2 | Gojo | 599 |

| 3 | Armin | 597 |

| 4 | Hisoka | 600 |

+------------+--------+------------+

**PRACTICAL:-29 Date:-**

# Refer Practical no 25 for table and Database :-

**Objective:**

**A student table contains roll number, name and age of all the students of a class. Write python program (using SQL query) to search an entered name into the table.**

**ALGORITHM:**

1. Import connector from mysql module .
2. Make the connection with the database .
3. Setup the cursor .
4. Take a input and store in variable (here search)
5. The execute the following query “select \* from students\_table WHERE ( id = '{search}'

OR Name= '{search}' OR age = '{search}' OR class = '{search}')"

1. Print all the results .

**CODE:**

**>>**import mysql.connector

**>>**a = mysql.connector.connect ( host='localhost' , user='root' , passwd='password', database='students\_details')

**>>**b = a.cursor()

**>>**name = input(' Enter student\'s name - ')

**>>**b.execute (f " SELECT \* FROM students\_details.name where Name = '{name}' ") **>>**x = b.fetchall() print(x)

**OUTPUT:**

Enter student's name - Armin Arlert

[ (6 , ' Armin Arlert ' , '16' ) ]

**PRACTICAL:-24 Date:-**

**Objective:**

**Write python program to read data from a CSV file and write its content in a table using SQL queries.**

**ALGORITHM :**

1. Import CSV and MySQL connector.
2. Open the CSV file and store the dates in a variable.
3. Connect my phone to MySQL server.
4. Please a cursor in the MySQL connection
5. Execute the query with the help of cursor.

**CODE:**

**i**mport csv

import mysql.connector

conn = mysql.connector.connect(user='user',password='password',  
 host='host', database='database')

cursor = conn.cursor()

with open('data.csv', 'r') as file:

reader = csv.reader(file)

header = next(reader)

# Create a table in the database

query = 'CREATE TABLE IF NOT EXISTS data ({} INT, {}  
 VARCHAR(255), {} INT)'.format(header[0], header[1],  
 header[2])

cursor.execute(query)

# Insert data into the table

for data in reader:

query = 'INSERT INTO data ({}, {}, {}) VALUES({}, "{}",  
 {})'.format(header[0], header[1], header[2], data[0],  
 data[1], data[2])

cursor.execute(query)

conn.commit()

conn.close()

**PRACTICAL:-30 Date:-**

**Objective:  
Create a new table (name, date of birth) by joining two tables (student id, name) and (student id, date of birth). Display all the details available in new table.**

**ALGORITHM :**

1. Use the database by any name.
2. Create two table,table1 with name student ID as parameter an table2 with student ID and date of birth.
3. Join the table by using UNION.
4. Now create another table by using SQL query with name and date of birth from the join table.

**CODE:**

USE DB12

Create table tb1 (Name (VAR) NOT NULL,student\_id (INT(50)) NOT NULL Primary Key);

Create table tb2 (student\_id (INT(50)) NOT NULL Primary Key,Date\_of\_Birth (VAR(50)Not Null);

CREATE TABLE newtable  
SELECT\*from tb1  
UNIOJN  
SELECT\*from tb2

CREATE TABLE newtable  
SELECT Name,Date\_of\_Birth

from tb1,tb2 WHERE tb1.student\_id=tb2.student\_id

**OUTPUT:**

|  |  |
| --- | --- |
| student\_id | Name |
| 101 | Rakesh |
| 103 | Anuj |
| 104 | Ram |

|  |  |
| --- | --- |
| student\_id | Date\_of\_Birth |
| 105 | 15/09/2004 |
| 101 | 05/07/2004 |
| 104 | 17/12/2005 |

|  |  |
| --- | --- |
| Name | Date\_of\_Birth |
| Rakesh | 05/07/2004 |
| Ram | 17/12/2005 |

**PRACTICAL:-31 Date:-**

**Objective:  
Create a new table (order ID, customer Name, and order Date) by joining two tables (order ID, customer ID, and order Date) and (customer ID, customer Name, contact Name, country).**

**ALGORITHM :**

1. Select a database.
2. Create two table with two different parameters.
3. Join the table using UNION.
4. Now create another table which contain (order ID, customer name and order table) from the tables using SQL queries.

**CODE:**

USE db\_12  
CREATE table1(OrderID VAR(50) NOT NULL , CustomerID  
 VAR(50) NOT NULL PRIMARY KEY, Orderdate  
 VAR(50) NOT NLL)

CREATE table2(CustomerID VAR(50) NOT NULL, CustomerName   
 VAR(50) NOT NULL , Country VAR(50) NOT NULL)

#Main table   
Select OrderID, CustomerName, Orderdate from table1,table2 WHERE table1.customer ID= table2.customer ID

**OUTPUT:  
 Table1**

|  |  |  |
| --- | --- | --- |
| OrderID | CustomerID | Orderdate |
| 105 | 10125 | 15/12/2023 |
| 107 | 10473 | 16/12/2023 |
| 102 | 10321 | 17/01/2023 |

**Table 2**

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | Contact | CustomerName | Country |
| 103 | 9876543211 | Rakesh | India |
| 102 | 8765432191 | Tom | USA |
| 107 | 8547921851 | Meena | India |

Main table

|  |  |  |
| --- | --- | --- |
| OrderID | CustomerName | Orderdate |
| 102 | Tom | 17/01/2023 |
| 107 | Meena | 16/12/2023 |

**PRACTICAL:-32 Date:-**

**Objective:  
A student table contains roll number, name and age of all the students of a class. Write python program (using SQL aggregate functions) to print min, max, sum and average age of the student in the table.**

**ALGORITHM :**

1. Use the database where the table is kept.
2. Connect the database to the Python.
3. Import MySQL.connector
4. Put a cursor to execute the query.
5. Fetch the data and print it.

**CODE:**

import mysql.connector

con = mysql.connector.connect(user='root',password='root',  
 host='localhost', database='db12')

cr=con.cursor()

cr.execute(SELECT \* MIN(AGE) from tb12)

f1=cr.fetchone()

print(f1)

cr1=con.cursor()

cr1.execute(SELECT \* MAX(AGE) from tb12)

f2=cr.fetchone()

print(f2)

cr2=con.cursor()

cr2.execute(SELECT \* AVG(AGE) from tb12)

f3=cr.fetchone()

print(f3)

cr3=con.cursor()

cr3.execute(SELECT \* SUM(AGE) from tb12)

f4=cr.fetchone()

print(f4)

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| Roll\_No | Name | Age |
| 1 | Tom | 17 |
| 2 | Gojo | 17 |
| 3 | B2K | 18 |
| 7 | Rakesh | 16 |

16

18

17

68

**PRACTICAL:-33 Date:-**

**Objective:  
An employee table contains Employee\_ID, name, age and salary of employees of a company. Write python program (using SQL query) to for the following-**

1. **Display name in descending order whose age is more than 40.**
2. **Display the average salary grouped by age.**
3. **Display the details of employee where salary is between 50000 and 80000.**
4. **Display the sum of salaries of employees whose name contains “sha”.**
5. **Display all the details of employees group by age having count(age)>3**

**ALGORITHM :**

1. Use the database where the table is kept.
2. Connect the database to Python.
3. Import the mySQL.connector
4. Put a cursor to execute the query.
5. Fetch and print the data.

**CODE:**

import mysql.connector

con = mysql.connector.connect(user='root',password='root',  
 host='localhost', database='db12',table=’tb12’)

cr=con.cursor()

cr.execute(SELECT NAME from tb12 WHERE AGE>40 ORDER BY

NAME,desc)

cr1=con.cursor()

cr1.execute(SELECT AVG(SALARY) from tb12 Group by AGE)

cr2=con.cursor()

cr2.execute(SELECT \* from tb12 WHERE

SALARY between 50000 and 80000)

cr3=con.cursor()

cr3.execute(SELECT SUM(SALARY)

from tb12 WHERE NAME = sha% )

cr4=con.cursor()

cr4.execute(SELECT \* from tb12 Group by AGE Having COUNT(AGE)>3)

**OUTPUT:**

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | NAME | AGE | SALARY |
| 1012 | Shayam | 45 | 60000 |
| 1031 | Ram | 47 | 65000 |
| 1047 | Shao | 45 | 70000 |
| 1092 | Rakesh | 47 | 50000 |
| 1056 | Ankit | 51 | 83000 |
| 1071 | Nikhil | 49 | 65000 |
| 1081 | Sachin | 49 | 55000 |
| 1095 | Shankar | 42 | 45000 |

**a) b)**

|  |
| --- |
| Name |
| Shayam |
| Ram |
| Ankit |
| Nikhil |
| Shankar |

|  |  |
| --- | --- |
| AGE | AVG(SALARY) |
| 45 | 62500 |
| 47 | 5750 |
| 49 | 60000 |

**b)**

|  |  |  |  |
| --- | --- | --- | --- |
| EMPLOYEE\_ID | NAME | AGE | SALARY |
| 1012 | Shayam | 45 | 60000 |
| 1031 | Ram | 47 | 65000 |
| 1047 | Shao | 45 | 70000 |
| 1081 | Sachin | 49 | 55000 |
| 1071 | Nikhil | 49 | 65000 |