**Q1. Write a program to print Floyd’s Triangle :**

**Code :**

**a=int(input("Enter the number of rows "))**

**k=0**

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

**for j in range (0,i):**

**k+=1**

**print(k, end=" ")**

**print()**

**Q2. Write a program to print Trapezium Pattern.**

**Code :**

num=int(input("Enter number of rows : "))

lterm = 1

rterm = num \* num + 1

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

for space in range(num, i-1, -1):

print(" ", end =" ")

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

print(str(lterm)+"\*", end ="")

lterm += 1

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

print(rterm, end ="")

if j < i:

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

rterm += 1

rterm = rterm - (i - 1) \* 2 - 1

print()

**Q3. Write a python program to find the GCD of two numbers.**

**Code :**

**import math**

**num1=int(input("Enter 1st number : "))**

**num2=int(input("Enter 2nd number : "))**

**print("The gcd of",num1,"and",num2,"is : ", end="")**

**print(math.gcd(num1, num2))**

**Without using import math :**

**def hcf(a, b):**

**if(b == 0):**

**return a**

**else:**

**return hcf(b, a % b)**

**a = int(input("Enter 1st number : "))**

**b = int(input("Enter 2nd number : "))**

**print("The gcd of",a,"and",b,"is : ", end="")**

**print(hcf(a,b))**

**Q4. Write a program to print the Fibonacci series.**

**Code :**

**nterms = int(input("Enter number of terms "))**

**n1=0**

**n2=1**

**count = 0**

**if nterms <= 0:**

**print("Please enter a positive integer")**

**elif nterms == 1:**

**print("Fibonacci sequence upto",nterms,":")**

**print(n1)**

**else:**

**print("Fibonacci sequence:")**

**while count < nterms:**

**print(n1,end=" ")**

**nth = n1 + n2**

**# update values**

**n1 = n2**

**n2 = nth**

**count += 1**

**Q5. Write a program to maintain names and cell numbers of 4 persons and then print them systematically in a tabular form.**

**Code :**

num = int(input("Enter the number of people : "))

for i in range (0,num):

s=input("Enter your name : ")

ph=int(input("Enter you phone number : "))

for i in range (0,num):

print(s,":",ph)

**Q6 . Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = ( 1 \* 1 \* 1 ) + ( 5 \* 5 \* 5 ) + ( 3 \* 3 \* 3 ).**

**Code :**

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

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

for num in range(lower,upper + 1):

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

if num == sum:

print(num)

**Q7. Write a Python program to convert two lists into a dictionary in a way that item from list1 is the key and item from list2 is the values.**

**Code :**

dic={}

lst1=[1,2,3,4]

lst2=['Jimin' ,'V','Jk','RM']

print("List one is ",lst1)

print("List 2 is ",lst2)

print("\n")

print("The combined list is ")

for i in lst1:

for j in lst2:

dic[i]=j

lst2.remove(j)

break

print (dic)

print("\n\n")

**Q8. Write a program to find all occurrences of 'T' in the string 'The Terrible Tiger Tore The Towel'. Replace all occurrences of 'T' with 't'.**

**Code :**

s = 'The Terrible Tiger Tore The Towel'

l=len(s)

for i in range (0,l):

if (s[i]=='T'):

print("T exists at position ",i)

print()

print("After replacement the statement is: ")

s=s.replace('T','t')

print(s)

**Book Code :**

s = 'The Terrible Tiger Tore The Towel'

pos = s.find('T', 0)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos, s[pos])

pos = s.find('T', pos + 1)

print(pos)

c = s.count('T')

s = s.replace('T', 't', c)

print(s)

**Q9. Write a Python program to create a dictionary storing shopping items in cart with details such as  ‐ (item name: cost of item) and perform the following operations**

1. Add a new item
2. Display item value greater than 100
3. Remove the details of the specific item

Removes the last item details from cart.

**Code :**

items = {}

#n = int(input("Total number of items in cart: "))

def add():

item\_name = input("Enter Name of Item: ")

value = int(input("Value of Item: "))

items.update({item\_name: value})

def d100():

for x,y in items.items():

if y>100:

print(x+" : ",y)

#print("Updated Cart: ",display())

def display():

print("ITEMS IN THE CART")

for x,y in items.items():

print(x+" : ",y)

def deleteitem():

display()

item\_delete = input("Enter Item name to delete: ")

items.pop(item\_delete)

print("Updated Cart: ")

display()

def delete\_last():

display()

items.popitem()

print("Updated Cart: ")

display()

c = 1

while(c==1):

print()

print()

print("-----MENU-----")

print("1.Add Item\n2.Display item value > 100\n3.Delete item\n4.Delete last item\n5.Display Items\n6.Exit")

ch = int(input("Enter Choice: "))

if ch==1:

add()

elif ch==2:

d100()

elif ch==3:

deleteitem()

elif ch==4:

delete\_last()

elif ch==5:

display()

elif ch==6:

c = 2

else:

print("Enter Correct value!!")

**Q10. Perform the following operations on a list of numbers. –**

1. Create a list of 5 odd numbers
2. Create a list of 5 even numbers
3. Combine the two lists
4. Add prime numbers 11, 17, 29 at the beginning of the combined list
5. Report how many elements are present in the list
6. Replace last 3 numbers in the list with 100, 200, 300
7. Reverse the list.
8. Delete the list

Code :

even\_list = []

odd\_list = []

ch =1

n = 5

for i in range(n):

ch=1

while(ch==1):

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

if num%2==0:

even\_list.append(num)

ch = 2

else:

print("Entered num is not even!")

ch = 1

for j in range(n):

c=1

while(c==1):

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

if num%2!=0:

odd\_list.append(num)

c = 2

else:

print("Entered num is not Odd!")

c = 1

combine\_list = even\_list + odd\_list

print("Combined List: ", combine\_list)

combine\_list.insert(0, 11)

combine\_list.insert(1, 17)

combine\_list.insert(2, 29)

print("Combined List: ", combine\_list)

print("Total number of elements: ", len(combine\_list))

print("Replacing Last 3 elemets with 100,200,300: ")

combine\_list.pop(len(combine\_list)-1)

combine\_list.pop(len(combine\_list)-1)

combine\_list.pop(len(combine\_list)-1)

combine\_list.append(100)

combine\_list.append(200)

combine\_list.append(300)

print("Updated List: ", combine\_list)

combine\_list.reverse()

print("Reverse list: ", combine\_list)

print("Removing the List.. ", combine\_list.clear())

**Q11. Write a Python function to  create user module to receive five integers from keyboard and get their sum and product calculated.**

**Code :**

def cal\_sum\_prod(v,w,x, y, z) :

ss = v+w+x + y + z

pp = v\*w\*x \* y \* z

return ss, pp

a = int(input('Enter a: '))

b = int(input('Enter b: '))

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

d = int(input('Enter d: '))

e = int(input('Enter e: '))

s, p = cal\_sum\_prod(a, b, c,d,e)

print("Sum is : ",s)

print("Product is :",p)

**Q12. A list contains tuples containing roll number, names and age of student. Write a Python program to gather all the names from this list into another list.**

**Code :**

lst = [('A101', 'Steve', 23), ('A104', 'Natasha', 25), ('A111', 'Tony', 24)]

nlst = [ ]

for ele in lst:

nlst = nlst + [ele[1]]

print(nlst)

**Q13. Write a necessary program to build a database connectivity for simple student and Mini Project management System using Python.**

1. Accept [Mini project name, Team members, Technologies used, Mini Project completion deadline]
2. Display ‐ displays the details of every Project
3. Search ‐ Look for a specific student and the project on which he or she is working
4. Delete ‐ delete a particular record (based on either project/employee

**Code :**

import pymysql

def get\_connection():

connection=pymysql.connect(host= 'localhost', user= 'root', password='sqlrocks@123', database='project\_management')

return connection

def accept(id, name, team\_members, technology, deadline):

connection=get\_connection()

cursor=connection.cursor()

query='''insert into projectinfo values ({},\'{}\',\'{}\',\'{}\',\'{}\')'''.format(id, name, team\_members, technology, deadline)

cursor.execute(query)

connection.commit()

print("Record Added Successfully")

def display():

connection= get\_connection()

cursor= connection.cursor()

query='''select \* from projectinfo;'''

cursor.execute(query)

data= cursor.fetchall()

for i in data:

print(i)

display()

def search(project\_name):

connection = get\_connection()

cursor = connection.cursor()

query='''select \* from projectinfo where name = \'{}\''''.format(project\_name)

cursor.execute(query)

data=cursor.fetchone()

print(data)

def update(team\_members):

connection = get\_connection()

cursor = connection.cursor()

query='''update projectinfo set members = \'{}\' '''.format(team\_members)

cursor.execute(query)

connection.commit()

update("Dhanlekshmi Yedurkar")

def delete(team\_members):

connection = get\_connection()

cursor = connection.cursor()

query='''delete from projectinfo where members = \'{}\''''.format(team\_members)

cursor.execute(query)

connection.commit()

delete("Dhanlekshmi Yedurkar")

**Q14. Write a program to build a simple Employee and Project management System using Python functions which can perform the following operations:**

1. Accept [Project name, Team members, Technologies used, Project completion deadline]
2. Display ‐ displays the details of every Project
3. Search ‐ Look for a specific employee and the project on which he or she is working
4. Delete ‐ delete a particular record (based on either project/employee

**Code :**

from ast import Delete

from dis import dis

projectname = []

empname = []

techused = []

completion\_date = []

def accept():

projectname.append(input("Enter Project name: "))

empname.append(input("Enter Employee Name along with His/Hers project name (projectname\_empname): "))

techused.append(input("Enter Tech Used along with project name (projectname\_techused): "))

completion\_date.append(input("Enter Project's Deadline(projectname\_deadline): "))

def display():

print("Projects: ", projectname)

for i in projectname:

print("Project name:", i)

print("Employees: ")

for j in empname:

check = 0

if i in j:

for w in j:

if w=="\_":

check = 1

continue

if check == 1:

print(w, end='')

print()

print("Tech Used: ")

for k in techused:

# print()

check = 0

if i in k:

for w in k:

if w=="\_":

check =1

continue

if check == 1:

print(w, end='')

print()

print("Deadline: ")

for l in completion\_date:

#print()

check = 0

if i in l:

for w in l:

if w=="\_":

check =1

continue

if check == 1:

print(w, end='')

print("----------")

print()

""" print("Employee names: ", empname)

print("Tech Used: ", techused)

print("Projects Deadline: ", completion\_date) """

def search():

print("Enter name of the employee to search: ")

emp\_name = input()

for i in empname:

if emp\_name in i:

print("Project Name: ",end='')

for j in i:

if (j=='\_'):

break

else:

print(j, end='')

else:

print("Employee not found!")

def delete():

print("Press 'e' to remove Employee or 'p' to remove Project: ")

z = input()

if (z=='p' or z=="P"):

print("Enter Project name to delete: ")

pro\_name = input()

for i in projectname:

if i==pro\_name:

projectname.remove(pro\_name)

print("Updated List of projects: ", projectname)

for i in empname:

if pro\_name in i:

empname.remove(i)

print("Updated List of employees: ", empname)

for i in techused:

if pro\_name in i:

techused.remove(i)

print("Updated List of techused: ", techused)

for i in completion\_date:

if pro\_name in i:

completion\_date.remove(i)

print("Updated List of Deadlines: ", completion\_date)

elif (z=='e' or z=='E'):

emp\_name = input("Enter Employee name to delete: ")

for i in empname:

if emp\_name in i:

empname.remove(i)

print("Updated List of employees: ", empname)

def update():

print("Enter Project name of the Employee: ")

project\_name = input()

print("Enter Name of the employee: ")

emp\_name = input()

empname.append(project\_name+"\_"+emp\_name)

print("Updated List of employees: ", empname)

x=1

while(x ==1 ):

print()

print("-----MENU-----")

print("1. Enter Data")

print("2. Display")

print("3. Search")

print("4. Delete(Project or Emp)")

print("5. Update employee")

print("6. Exit")

print("Enter Choice: ")

c = int(input())

if (c==1):

accept()

elif (c==2):

display()

elif (c==3):

search()

elif (c==5):

update()

elif (c==4):

delete()

elif (c==6):

        break

**Q16. Write a program to read the contents of file ‘message’ one character at a time. Print each character that is read.**

**Code :**

from time import sleep

with open('myfile.txt', 'r') as fp:

for letter in iter(lambda: fp.read(1), ''):

print(letter)

      sleep(0.1)

**Q15. Write a python program to demonstrate different functions of OS module in python.**

**Code :**

import os

cwd = os.getcwd()

print("Current working directory: ",cwd)

#Listing out Files and Directories with Python

path ='C:\MIT ADT\Second year - Sem - 1\IPCL'#use '/' for root folder

dir\_list = os.listdir(path)

print("Files and directories in '", path, "' :")

print(dir\_list)

print()

#This method will check whether a file exists or not

result = os.path.exists('13\_sql.py')

print(result)

#python will give us the size of the file in bytes.

print("Size: ",os.path.getsize('11\_sumprod.py'))

#rename file if it exists os.rename('oldname', 'newname')

os.rename('temp.py', 'temp1.py')

#remove file using os.remove('filename')

os.remove('temp2.py')

**Q17. Write a Python program that accepts a list of integers and checks the length and the third element. Return true if the length of the list is more than 10 and the third element occurs twice in the said list. And also Sort the list in ascending order.**

**Code :**

l1=[]

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

for i in range(0,n):

e=int(input("Enter the elements"))

l1.append(e)

print("The original list is ",l1)

l2=l1

l2.remove(l1[2])

a=len(l1)

print("The elements after removing the 3rd term ")

print(l2)

print("The lenth of the list is ",a)

if(l2[2] in l1 and a>10):

print ("There are more than 10 elements and the 3rd element is repeated ")

else:

print("The list doesnt contain more than 10 elements")

print("\*"\*80)

l2.sort()

print("Sorted list :",l2)

**Q18. Write a generic program to handle exception generated in following scenarios:**

1. Division by Zero
2. Accessing a file which does not exist.
3. Addition of two incompatible types
4. Trying to access a nonexistent index of a sequence"

**Code :**

print("Exception for division by zero")

a=5

b=0

try:

print("The result of division is : ", a/b)

except ZeroDivisionError:

print("Division by 0 is not allowed")

print()

print("Accesing file that doesnt exist")

try:

open('new.txt','r')

except FileNotFoundError:

print("File does not exist")

print()

print("Addition of 2 incompatable data types ")

x=8

y='d'

try:

print("The addition is : ", x+y)

except TypeError:

print("The values are of different types")

print()

print("Accesing a non existant index statement ")

list1=[1,2,3,'animus']

try:

print("The seventh value is : ", list1[6])

except IndexError:

print("The index is out of range")

**Q19. Write a program that infinitely receives positive integer as input and prints its square. If a negative number is entered then raise an exception, display a relevant error message and make a graceful exit.**

**Code :**

try:

while True :

num = int(input('Enter a positive number: '))

if num >= 0 :

print(num \* num)

else :

print('Negative number')

except ValueError as ve :

  print(ve.args)

**Q20. Write a Python program that reports the time of creation, time of last access and time of last modification for a given file.**

**Code :**

import os

import time

path = "myfile.txt"

ti\_c = os.path.getctime(path)

ti\_m = os.path.getmtime(path)

c\_ti = time.ctime(ti\_c)

m\_ti = time.ctime(ti\_m)

print("The file located at the path ",{path} )

print("The file was created at ",{c\_ti} )

print("The file was last modified at ",{m\_ti})

**Q21. Write a program that receives an integer as input. If a string is entered instead of an integer, then report an error and give another chance to user to enter an integer. Continue this process till correct input is supplied.**

**Code :**

while True :

try :

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

break

except ValueError :

print('Incorrect Input')

print('You entered: ', num)