**1a. Write a python program to find the best of two test average marks out of three test’s marks accepted from the user.**

m1 = int (input("Enter the marks in the first test: "))

m2 = int (input("Enter the marks in second test: "))

m3 = int (input("Enter the marks in third test: "))

if (m1 > m2):

if (m2 > m3):

total = m1 + m2

else:

total = m1 + m3

elif (m1 > m3):

total = m1 + m2

else:

total = m2 + m3

Avg = total / 2

print ("The average of the best two test marks is: ",Avg)

**Output:**

**Case 1:**

Enter the marks in the first test: 20

Enter the marks in the second test: 15

Enter the marks in the third test: 22

The average of the best two test marks is: 21.0

**Case 2:**

Enter the marks in the first test: 20

Enter the marks in the second test: 23

Enter the marks in the third test: 18

The average of the best two test marks is: 21.5

OR

m1 = int(input("Enter marks for test1 : "))

m2 = int(input("Enter marks for test2 : "))

m3 = int(input("Enter marks for test3 : "))

if m1 <= m2 and m1 <= m3:

Avg = (m2+m3)/2

elif m2 <= m1 and m2 <= m3:

Avg = (m1+m3)/2

elif m3 <= m1 and m2 <= m2:

Avg = (m1+m2)/2

print("Average of best two test marks out of three test’s marks is", Avg);

**1b. Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number**

Number = int(input("Please Enter any Number: "))

Reverse = 0

temp=Number

while(Number > 0):                         #121>0 12>0

  Reminder = Number %10             # rem=121%10= 1    rem=12%10=2

  Reverse = (Reverse \*10) + Reminder     # rev=0\*10+1=1 1\*10+2=12

  Number = Number //10                     # num=121//10=12 12//10=1

print("\n Reverse of entered number is",Reverse) # rem=1%10=1 rev=121 n=0

if  temp == Reverse:

  print("The given number {0} is palindrome".format(temp))

else:

  print("The given number {0} is not palindrome".format(temp))

**Output:**

**Case1:**

Please Enter any Number: 121

Reverse of entered number is 121

The given number 121 is palindrome

**Case2:**

Please Enter any Number: 123

Reverse of entered number is 321

The given number 123 is not palindrome

Number = int(input("Please Enter any Number: "))

digit = int(input("Please Enter digit to find the occurrence: "))

count=0

while (Number > 0):

        # check if the digit is D          #n=1231 d=1

        if(Number % 10 == digit):   #1231%10=1 1==1 123%10=3 3==1 12%10=2

            count=count+1                                                #count=0+1=1

         Number = Number // 10         # num=1231//10=123 123//10=12

print(" digit {0} occurs {1} times ".format(digit,count))

**Output**

**Case1:**

Please Enter any Number: 123121

Please Enter digit to find the occurrence: 1

digit 1 occurs 3 times

**Case2:**

Please Enter any Number: 12277189292777456

Please Enter digit to find the occurrence: 7

digit 7 occurs 5 times

val = int(input("Enter a value : "))

str\_val = str(val)

if str\_val == str\_val[::-1]:

print("Palindrome")

else:

print("Not Palindrome")

for i in range(10):

if str\_val.count(str(i)) > 0:

print(str(i),"appears", str\_val.count(str(i)), "times");

**2a. Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function*.* Display suitable error message if the condition for input value is not followed.**

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

Fn = Fn-1 + Fn-2

With seed values

F0 = 0 and F1 = 1.

# Function for nth fibonacci number - Space Optimisation

# Taking 1st two fibonacci numbers as 0 and 1

**def** fibonacci(n):

    a **=** 0

    b **=** 1

**if** n < 0:

**print**("Incorrect input")

**elif** n **==** 0:

**return** a

**elif** n **==** 1:

**return** b

**else**:

**for** i **in** range(2, n):

            c **=** a **+** b

            a **=** b

            b **=** c

**return** b

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

print(fibonacci(num))

# Function for nth Fibonacci number

def Fibonacci(n):

if n<= 0:

print("Incorrect input")

# First Fibonacci number is 0

elif n == 1:

return 0

# Second Fibonacci number is 1

elif n == 2:

return 1

else:

return Fibonacci(n-1)+Fibonacci(n-2)

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

print(Fibonacci(num))

**2b. Develop a python program to convert binary to decimal, octal to hexadecimal using functions**

|  |
| --- |
| # Function calculates the decimal equivalent  # to given binary number      **def** binaryToDecimal(binary):        decimal, i **=** 0, 0  **while**(binary !**=** 0): #100          dec **=** binary **%** 10 #dec=100%10=0          decimal **=** decimal **+** dec **\*** pow(2, i) #decimal=0+0\*pow(2,0)=0          binary **=** binary**//**10 #bin=100//10= 10          i **+=** 1 #i=1  **print**(decimal)  # dec=10%10=0  #decimal=0+0\*pow(2,1)=0  # Bin=10//10=1, i=2  num = int(input("Enter a binary number : "))  print(binaryToDecimal (num))      #dec=1%10=1       #decimal=0+1\*pow(2,2)=4      #bin=1//10=0 i=3 |
|  |

2b. Octal to Hexadecimal

def oct2Hex(val):

    rev=val[::-1]

    dec = 0

    i = 0

    for dig in rev:

        dec += int(dig) \* 8\*\*i

        i += 1

    list=[]

    while dec != 0:

        list.append(dec%16)

        dec = dec // 16

    nl=[]

    for elem in list[::-1]:

        if elem <= 9:

            nl.append(str(elem))

        else:

            nl.append(chr(ord('A') + (elem -10)))

    hex = "".join(nl)

    return hex

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

print(oct2Hex(num))

def fn(n):

    if n == 1:

        return 0

    elif n == 2:

        return 1

    else:

        return fn(n-1) + fn(n-2)

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

if num > 0:

    print("fn(", num, ") = ",fn(num) , sep ="")

else:

    print("Error in input")

**3a.Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 22 01:37:07 2023

......................

"""

sentence = input("Enter a sentence : ")

wordList = sentence.split(" ")

print("This sentence has", len(wordList), "words")

digCnt = upCnt = loCnt = 0

for ch in sentence:

if '0' <= ch <= '9':

digCnt += 1

elif 'A' <= ch <= 'Z':

upCnt += 1

elif 'a' <= ch <= 'z':

loCnt += 1

print("This sentence has", digCnt, "digits", upCnt, "upper case letters", loCnt, "lower case letters")

Output

Enter a sentence : Rama went to Devaraja market to pick 2 kgs of vegetable

This sentence has 11 words

This sentence has 1 digits 2 upper case letters 42 lower case letters

**String Similarity**

**3b.Write a Python program to find the string similarity between two given strings.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Wed Feb 22 01:48:17 2023

......................

"""

str1 = input("Enter String 1 \n")

str2 = input("Enter String 2 \n")

if len(str2) < len(str1):

short = len(str2)

long = len(str1)

else:

short = len(str1)

long = len(str2)

matchCnt = 0

for i in range(short):

if str1[i] == str2[i]:

matchCnt += 1

print("Similarity between two said strings:")

print(matchCnt/long)

Output

Enter String 1

Python Exercises

Enter String 2

Python Exercises

Similarity between two said strings:

1.0

Enter String 1

Python Exercises

Enter String 2

Python Exercise

Similarity between two said strings:

0.9375

**Question 4**

**4a.Insertion Sort & Merge Sort on lists**

**Write a python program to implement insertion sort and merge sort using lists.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 04:06:09 2023

......................

"""

import random

def merge\_sort(lst):

if len(lst) > 1:

mid = len(lst) // 2

left\_half = lst[:mid]

right\_half = lst[mid:]

merge\_sort(left\_half)

merge\_sort(right\_half)

i = j = k = 0

while i < len(left\_half) and j < len(right\_half):

if left\_half[i] < right\_half[j]:

lst[k] = left\_half[i]

i += 1

else:

lst[k] = right\_half[j]

j += 1

k += 1

while i < len(left\_half):

lst[k] = left\_half[i]

i += 1

k += 1

while j < len(right\_half):

lst[k] = right\_half[j]

j += 1

k += 1

return lst

def insertion\_sort(arr):

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

key = arr[i]

j = i - 1

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

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

my\_list = []

for i in range(10):

my\_list.append(random.randint(0, 999))

print("\nUnsorted List")

print(my\_list)

print("Sorting using Insertion Sort")

insertion\_sort(my\_list)

print(my\_list)

my\_list = []

for i in range(10):

my\_list.append(random.randint(0, 999))

print("\nUnsorted List")

print(my\_list)

print("Sorting using Merge Sort")

merge\_sort(my\_list)

print(my\_list)

Output

Unsorted List

[932, 111, 226, 685, 543, 589, 918, 539, 294, 717]

Sorting using Insertion Sort

[111, 226, 294, 539, 543, 589, 685, 717, 918, 932]

Unsorted List

[613, 176, 828, 265, 65, 326, 359, 919, 514, 868]

Sorting using Merge Sort

[65, 176, 265, 326, 359, 514, 613, 828, 868, 919]

**Roman to Integer Conversion**

**4b.Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-"""

Created on Wed Feb 22 02:05:25 2023

......................

"""

def roman2Dec(romStr):

roman\_dict ={'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}

# Analyze string backwards

romanBack = list(romStr)[::-1]

value = 0

# To keep track of order

rightVal = roman\_dict[romanBack[0]]

for numeral in romanBack:

leftVal = roman\_dict[numeral]

# Check for subtraction

if leftVal < rightVal:

value -= leftVal

else:

value += leftVal

rightVal = leftVal

return value

romanStr = input("Enter a Roman Number : ")

print(roman2Dec(romanStr))

Output

Enter a Roman Number : XVII

17

Enter a Roman Number : MLXVI

1066

**Question 5**

**5a. Check Phone Number**

**Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 04:19:57 2023

......................

"""

import re

def isphonenumber(numStr):

if len(numStr) != 12:

return False

for i in range(len(numStr)):

if i==3 or i==7:

if numStr[i] != "-":

return False

else:

if numStr[i].isdigit() == False:

return False

return True

def chkphonenumber(numStr):

ph\_no\_pattern = re.compile(r'^\d{3}-\d{3}-\d{4}$')

if ph\_no\_pattern.match(numStr):

return True

else:

return False

ph\_num = input("Enter a phone number : ")

print("Without using Regular Expression")

if isphonenumber(ph\_num):

print("Valid phone number")

else:

print("Invalid phone number")

print("Using Regular Expression")

if chkphonenumber(ph\_num):

print("Valid phone number")

else:

print("Invalid phone number")

Output

Enter a phone number : 444-654-5656

Without using Regular Expression

Valid phone number

Using Regular Expression

Valid phone number

Enter a phone number : 45A4-444-878

Without using Regular Expression

Invalid phone number

Using Regular Expression

Invalid phone number

**Search Phone Number & Email**

**5b.Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com)**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 04:40:10 2023

......................

"""

import re

# Define the regular expression for phone numbers

phone\_regex = re.compile(r'\+\d{12}')

email\_regex = re.compile(r'[A-Za-z0-9.\_]+@[A-Za-z0-9]+\.[A-Z|a-z]{2,}')

# Open the file for reading

with open('example.txt', 'r') as f:

# Loop through each line in the file

for line in f:

# Search for phone numbers in the line

matches = phone\_regex.findall(line)

# Print any matches found

for match in matches:

print(match)

matches = email\_regex.findall(line)

# Print any matches found

for match in matches:

print(match)

Output

+918151894220

+829392938876

+918768456234

prakash81.82@gmail.in

**Question 6**

**File Operations**

**6a. Write a python program to accept a file name from the user and perform the following operations**

**Display the first N line of the file**

**Find the frequency of occurrence of the word accepted from the user in the file**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 05:26:33 2023

......................

"""

import os.path

import sys

fname = input("Enter the filename : ")

if not os.path.isfile(fname):

print("File", fname, "doesn't exists")

sys.exit(0)

infile = open(fname, "r")

lineList = infile.readlines()

for i in range(20):

print(i+1, ":", lineList[i])

word = input("Enter a word : ")

cnt = 0

for line in lineList:

cnt += line.count(word)

print("The word", word, "appears", cnt, "times in the file")

Output

Enter the filename : example.txt

1 : this is phone number +918151894220

2 : no phone number here

3 : here we have one +829392938876

4 : we have an email prakash81.82@gmail.in and a number +918768456234

5 : nothing of that sort here

6 : Better hope the life-inspector doesn't come around while you have your

7 : life in such a mess.

8 : You can create your own opportunities this week. Blackmail a senior executive.

9 : Be different: conform.

10 : Be cheerful while you are alive.

11 : -- Phathotep, 24th Century B.C.

12 : Q: How many journalists does it take to screw in a light bulb?

13 : A: Three. One to report it as an inspired government program to bring

14 : light to the people, one to report it as a diabolical government plot

15 : to deprive the poor of darkness, and one to win a Pulitzer prize for

16 : reporting that Electric Company hired a light bulb-assassin to break

17 : the bulb in the first place.

18 : Q: Why did the astrophysicist order three hamburgers?

19 : A: Because he was hungry.

20 : Q: Why haven't you graduated yet?

Enter a word : the

The word the appears 7 times in the file

**Zip operation on a folder**

**6b.Develop a program to backing Up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Fri Dec 23 16:14:28 2022

......................

"""

import os

import sys

import pathlib

import zipfile

dirName = input("Enter Directory name that you want to backup : ")

if not os.path.isdir(dirName):

print("Directory", dirName, "doesn't exists")

sys.exit(0)

curDirectory = pathlib.Path(dirName)

with zipfile.ZipFile("myZip.zip", mode="w") as archive:

for file\_path in curDirectory.rglob("\*"):

archive.write(file\_path, arcname=file\_path.relative\_to(curDirectory))

if os.path.isfile("myZip.zip"):

print("Archive", "myZip.zip", "created successfully")

else:

print("Error in creating zip archive")

Output

Enter Directory name that you want to backup : zipDemo

Archive myZip.zip created successfully

**Question 7**

**Inheritance**

**7a.By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 05:40:37 2023

......................

"""

import math

class Shape:

def \_\_init\_\_(self):

self.area = 0

self.name = ""

def showArea(self):

print("The area of the", self.name, "is", self.area, "units")

class Circle(Shape):

def \_\_init\_\_(self,radius):

self.area = 0

self.name = "Circle"

self.radius = radius

def calcArea(self):

self.area = math.pi \* self.radius \* self.radius

class Rectangle(Shape):

def \_\_init\_\_(self,length,breadth):

self.area = 0

self.name = "Rectangle"

self.length = length

self.breadth = breadth

def calcArea(self):

self.area = self.length \* self.breadth

class Triangle(Shape):

def \_\_init\_\_(self,base,height):

self.area = 0

self.name = "Triangle"

self.base = base

self.height = height

def calcArea(self):

self.area = self.base \* self.height / 2

c1 = Circle(5)

c1.calcArea()

c1.showArea()

r1 = Rectangle(5, 4)

r1.calcArea()

r1.showArea()

t1 = Triangle(3, 4)

t1.calcArea()

t1.showArea()

Output

The area of the Circle is 78.53981633974483 units

The area of the Rectangle is 20 units

The area of the Triangle is 6.0 units

**Employee Details**

**7b. Write a python program by creating a class called Employee to store the details of Name, Employee\_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 12:09:50 2023

......................

"""

class Employee:

def \_\_init\_\_(self):

self.name = ""

self.empId = ""

self.dept = ""

self.salary = 0

def getEmpDetails(self):

self.name = input("Enter Employee name : ")

self.empId = input("Enter Employee ID : ")

self.dept = input("Enter Employee Dept : ")

self.salary = int(input("Enter Employee Salary : "))

def showEmpDetails(self):

print("Employee Details")

print("Name : ", self.name)

print("ID : ", self.empId)

print("Dept : ", self.dept)

print("Salary : ", self.salary)

def updtSalary(self):

self.salary = int(input("Enter new Salary : "))

print("Updated Salary", self.salary)

e1 = Employee()

e1.getEmpDetails()

e1.showEmpDetails()

e1.updtSalary()

Output

Enter Employee name : Sameer

Enter Employee ID : A123

Enter Employee Dept : CSE

Enter Employee Salary : 85750

Employee Details

Name : Sameer

ID : A123

Dept : CSE

Salary : 85750

Enter new Salary : 88800

Updated Salary 88800

**Question 8**

**Polymorphism and Inheritance**

**8a. Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 12:20:20 2023

......................

"""

class PaliStr:

def \_\_init\_\_(self):

self.isPali = False

def chkPalindrome(self, myStr):

if myStr == myStr[::-1]:

self.isPali = True

else:

self.isPali = False

return self.isPali

class PaliInt(PaliStr):

def \_\_init\_\_(self):

self.isPali = False

def chkPalindrome(self, val):

temp = val

rev = 0

while temp != 0:

dig = temp % 10

rev = (rev\*10) + dig

temp = temp //10

if val == rev:

self.isPali = True

else:

self.isPali = False

return self.isPali

st = input("Enter a string : ")

stObj = PaliStr()

if stObj.chkPalindrome(st):

print("Given string is a Palindrome")

else:

print("Given string is not a Palindrome")

val = int(input("Enter a integer : "))

intObj = PaliInt()

if intObj.chkPalindrome(val):

print("Given integer is a Palindrome")

else:

print("Given integer is not a Palindrome")

Output

Enter a string : madam

Given string is a Palindrome

Enter a integer : 567587

Given integer is not a Palindrome

Enter a string : INDIA

Given string is not a Palindrome

Enter a integer : 6789876

Given integer is a Palindrome

**Question 9**

**Download XKCD comics**

**9a. Write a python program to download the all XKCD comics**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 13:02:47 2023

......................

"""

import os

import requests

import io

from PIL import Image

# Define a function to download a single XKCD comic

def download\_comic(comic\_id, directory):

# Construct the URL for the XKCD API

url = f'https://xkcd.com/{comic\_id}/info.0.json'

# Make an HTTP GET request to the XKCD API

response = requests.get(url)

# Parse the JSON response

data = response.json()

# Extract the image URL from the data dictionary

image\_url = data['img']

# Make an HTTP GET request to the image URL

response = requests.get(image\_url)

# Open the image using Pillow

image = Image.open(io.BytesIO(response.content))

# Create the directory if it doesn't exist

directory="download\_comics"

os.makedirs(directory, exist\_ok=True)

# Save the image to a file in the specified directory

filename = f'{comic\_id}.png'

filepath = os.path.join(directory, filename)

image.save(filepath, 'PNG')

print(f'Saved {filename}')

# Define a function to download all XKCD comics

def download\_all\_comics(start\_id, end\_id, directory):

for comic\_id in range(start\_id, end\_id + 1):

try:

# Download the comic

download\_comic(comic\_id, directory)

except Exception as e:

print(f'Error downloading comic {comic\_id}: {e}')

# Call the download\_all\_comics function to download the first 10 XKCD comics

download\_all\_comics(1, 10, 'xkcd\_images')

**Output:**

Saved 1.png

Saved 2.png

Saved 3.png

Saved 4.png

Saved 5.png

Saved 6.png

Saved 7.png

Saved 8.png

Saved 9.png

Saved 10.png

**Spreadsheet Operations**

**9b. Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet**

**Python Code**

import openpyxl

# Read data from a spreadsheet

def read\_spreadsheet(file\_path, sheet\_name):

workbook = openpyxl.load\_workbook(file\_path)

sheet = workbook[sheet\_name]

data = []

for row in sheet.iter\_rows():

row\_data=[]

for cell in row:

row\_data.append(cell.value)

data.append(row\_data)

workbook.close()

return data

# Write data to a spreadsheet

def write\_spreadsheet(file\_path, sheet\_name, data):

workbook = openpyxl.load\_workbook(file\_path)

sheet = workbook[sheet\_name]

for row in data:

sheet.append(row)

workbook.save(file\_path)

workbook.close()

# Example usage

file\_path = 'Example.xlsx' # Replace with the path to your spreadsheet file

sheet\_name = 'Sheet1' # Replace with the name of your sheet

# Read data from the spreadsheet

read\_data = read\_spreadsheet(file\_path, sheet\_name)

print("Data read from spreadsheet:")

for row in read\_data:

print(row)

# Write data to the spreadsheet

new\_data = [

['John', 'Doe', 25],

['Jane', 'Smith', 30],

['Bob', 'Johnson', 35]

]

write\_spreadsheet(file\_path, sheet\_name, new\_data)

print("Data written to spreadsheet.")

**Output:**

Data read from spreadsheet:

['John', 'Doe', 25]

Data written to spreadsheet.

**Question 10**

**Merge selected pages from Multiple PDFs to a new PDF**

**10a. Write a python program to combine select pages from many PDFs**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 9 17:34:55 2023

......................

"""

from PyPDF2 import PdfWriter, PdfReader

num = int(input("Enter page number you want combine from multiple documents "))

pdf1 = open('birds.pdf', 'rb')

pdf2 = open('birdspic.pdf', 'rb')

pdf\_writer = PdfWriter()

pdf1\_reader = PdfReader(pdf1)

page = pdf1\_reader.pages[num - 1]

pdf\_writer.add\_page(page)

pdf2\_reader = PdfReader(pdf2)

page = pdf2\_reader.pages[num - 1]

pdf\_writer.add\_page(page)

with open('output.pdf', 'wb') as output:

pdf\_writer.write(output)

Output

This program allows you to extract specific pages from two PDF files, “birds.pdf” and “birdspic.pdf,” by entering the page numbers as user input. Once you input the desired page numbers, the program fetches those pages from both PDF files and combines them into a new file called “output.pdf.” This way, you can easily compile the desired pages from multiple PDF files into one document for your convenience.

Enter page number you want combine from multiple documents 3

[birdsDownload](https://moodle.sit.ac.in/blog/wp-content/uploads/2023/03/birds.pdf)

[birdspicDownload](https://moodle.sit.ac.in/blog/wp-content/uploads/2023/03/birdspic.pdf)

[outputDownload](https://moodle.sit.ac.in/blog/wp-content/uploads/2023/03/output.pdf)

**Fetch weather data from the JSON**

**10b. Write a python program to fetch current weather data from the JSON file**

**Python Code**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Fri Mar 10 09:22:53 2023

......................

"""

import json

# Load the JSON data from file

with open('weather\_data.json') as f:

data = json.load(f)

# Extract the required weather data

current\_temp = data['main']['temp']

humidity = data['main']['humidity']

weather\_desc = data['weather'][0]['description']

# Display the weather data

print(f"Current temperature: {current\_temp}°C")

print(f"Humidity: {humidity}%")

print(f"Weather description: {weather\_desc}")

JSON File :

{

"coord": {

"lon": -73.99,

"lat": 40.73

},

"weather": [

{

"id": 800,

"main": "Clear",

"description": "clear sky",

"icon": "01d"

}

],

"base": "stations",

"main": {

"temp": 15.45,

"feels\_like": 12.74,

"temp\_min": 14.44,

"temp\_max": 16.11,

"pressure": 1017,

"humidity": 64

},

"visibility": 10000,

"wind": {

"speed": 4.63,

"deg": 180

},

"clouds": {

"all": 1

},

"dt": 1617979985,

"sys": {

"type": 1,

"id": 5141,

"country": "US",

"sunrise": 1617951158,

"sunset": 1618000213

},

"timezone": -14400,

"id": 5128581,

"name": "New York",

"cod": 200

}

Output

Current temperature: 15.45°C

Humidity: 64%

Weather description: clear sky