

Lotus Valley International School



CS Practical File

Pranav Verma
XII Aryabhatta

Chapter 1

Review of Python Basics

Q1) Write a program to multiply an element by two, if it is an odd index for a given list containing both numbers and strings.

CODE

```
l=eval(input("Enter an Input: "))

n=len(l)

for i in range(n):

    if i%2!=0:

        l[i]=l[i]*2

print(l)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q18.py
Enter an Input: (1,"abc","hello_world_",4,7,9)
[1, 'abcabc', 'hello_world_', 8, 7, 18]
```

Q2) Write a program to count the frequency of an element in a list.

CODE

```
l=eval(input("Enter an Input: "))

e=eval(input('Enter the element to be counted: '))

print(l.count(e))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q19.py
Enter an Input: (1,2,1,3,4,5,6,1,3,4,4,2,5,6)
Enter the element to be counted: 1
3
```

Q3) Write a program to shift elements of a list so that the first element moves to the second index and second index move to the third index and so on, and the last element shifts to the first position.

Suppose the list is [10, 20, 30, 40].

After shifting, it should look like: [40, 10, 20, 30].

CODE

```
l=eval(input('Enter an Input: '))
x=l[-1]
l.pop(-1)
l.insert(0,x)
print(l)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q20.py
Enter an Input: (10,20,30,40)
(40, 10, 20, 30)
```

Q4) A list NUM contains the elements: 3, 25, 13, 6, 35, 8, 14, 45.

Write a program to swap the content with the next value divisible by 5 so that the resultant list will look like: 25, 3, 13, 35, 6, 8, 45, 14.

CODE

```
l=eval(input('Enter an Input: '))
n=len(l)
for i in range(n):
```

```
    if l[i]%5==0:
        l[i-1],l[i]=l[i],l[i-1]
```

```
print(l)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q21.py
Enter an Input: (3,25,13,6,35,8,14,45)
(25, 3, 13, 35, 6, 8, 45, 14)
```

Q5) Write a program to accept values from a user in a tube. Add a tuple to it and display its elements one by one. Also display its maximum and minimum values.

CODE

```
n=int(input('Enter No. of Elements: '))
```

```
t=()
for i in range(n):
    x=eval(input('Enter a Value: '))
    t+=x,
print(t)
for i in t:
    print(i)
print(max(t),min(t),end='\n')
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q22.py
Enter No. of Elements: 6
Enter a Value: 1
Enter a Value: 3
Enter a Value: 5
Enter a Value: 2
Enter a Value: 4
Enter a Value: 7
(1, 3, 5, 2, 4, 7)
1
3
5
2
4
7
7 1
```

Q6) Write a program to input any values for two tuples. Print it, interchange it and then compare them.

CODE

```
t1=eval(input("Enter the Input: "))
t2=eval(input("Enter the Input: "))
print(t1,t2)  x=t2  t2=t1  t1=x
print(t1,t2) if t1<t2:
```

```
print('t2 is greater')
```

else:

```
    print('t1 is greater')
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q23.py
Enter the Input: (1,3,5,2,4,7,9,8,0)
Enter the Input: (0,9,8,7,6,5,3,2,1)
(1, 3, 5, 2, 4, 7, 9, 8, 0) (0, 9, 8, 7, 6, 5, 3, 2, 1)
(0, 9, 8, 7, 6, 5, 3, 2, 1) (1, 3, 5, 2, 4, 7, 9, 8, 0)
t2 is greater
```

Q7) Write a python program to input 'n' classes and names of class teachers to store them in a dictionary and display the same. Also accept a particular class from the user and display the name of the class teacher of that class.

CODE

```
n=int(input('Enter No. of entries: '))
```

```
d={}
```

```
for i in range(n):
```

```
    x=input('Enter the Class name: ')
```

```
    y=input('Enter Class Teacher Name: ')
```

```
    d[x]=y
```

```
print(d)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q24.py
Enter No. of entries: 4
Enter the Class name: A
Enter Class Teacher Name: a
Enter the Class name: B
Enter Class Teacher Name: b
Enter the Class name: C
Enter Class Teacher Name: c
Enter the Class name: D
Enter Class Teacher Name: d
{'A': 'a', 'B': 'b', 'C': 'c', 'D': 'd'}
```

Q8) Write a program to store students names and their percentage in a dictionary and delete a particular student name from the dictionary. Also display the dictionary after deletion.

CODE

```
d=eval(input('Enter a Dictionary: '))
x=input('Enter the name of the student to be deleted: ')
if x in d:
    del d[x]
print(d)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q25.py
Enter a Dictionary: {"A":90,"B":92,"C":89,"D":96,"E":94,"F":93,"G":91,"H":95}
Enter the name of the student to be deleted: F
{'A': 90, 'B': 92, 'C': 89, 'D': 96, 'E': 94, 'G': 91, 'H': 95}
```

Q9) Write a Python program to input names of 'n' customers and their details like items bought, cost and phone number, etc., store them in a dictionary and display all the details in a tabular form.

CODE

```
d={}
n=int(input('Enter No. of Customers: '))
for i in range(n):
    name=input('Enter name of Customer: ')
    item=input('Enter item bought: ')
    cost=eval(input('Enter the cost of item: '))
    ph_no=int(input('Enter contact no.: '))
    d[name]=[item,cost,ph_no]

print('Name','\t','Item','\t','Cost','\t','Contact Number')
for i in d:
    print(i,' ',d[i][0],'\t',d[i][1],'\t',d[i][2])
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q26.py
Enter No. of Customers: 3
Enter name of Customer: A
Enter item bought: a
Enter the cost of item: 10
Enter contact no.: 123456
Enter name of Customer: B
Enter item bought: b
Enter the cost of item: 7.99
Enter contact no.: 243165
Enter name of Customer: C
Enter item bought: c
Enter the cost of item: 5.99
Enter contact no.: 327947
```

Name	Item	Cost	Contact Number
A	a	10	123456
B	b	7.99	243165
C	c	5.99	327947

Q10) Write a Python program to capitalize the first and last letters of each word of a given string.

CODE

```
s=input('Enter a string: ')
j=""
x=""
l=s.split()
for i in l:
    j=i[0].upper()+i[1:-1]+i[-1].upper()
    x+=j+' '
print(x)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q27.py
Enter a string: hello world
Hello World
```

Q11) Write a Python program to remove duplicate characters of a given string.

CODE

```
s=input('Enter a string: ')
```

```
x=""  
for i in s:  
    if i not in x:  
        x+=i  
print(x)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q28.py  
Enter a string: Hello World, Practical File Question 11 Test  
Hello Wrd,PractFQusn1T
```

Q12) Write a Python program compute the sum of digits of a given number

CODE

```
x=int(input('Enter a number: '))  
s=0  
while x>0:  
    b=x%10  
    s+=b  
    x=x//10  
print(s)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q29.py  
Enter a number: 1280  
11
```

Q13) Write a Python program to find the second most repeated word in a given string.

CODE

```
s = input("Enter a string: ")  
words = s.split()  
freq = {}  
for word in words:  
    freq[word] = freq.get(word, 0) + 1
```

```
sorted_freq = sorted(freq.items(), key=lambda x: x[1], reverse=True)
if len(sorted_freq) < 2:
    print("No second most repeated word found.")
else:
    print("Second most repeated word:", sorted_freq[1][0])
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q30.py
Enter a string: this is a test it is just a test test
Second most repeated word: is
```

Q14) Write a Python program to change a given string to a new string where the first and last string have been exchanged.

CODE

```
s=input('Enter a string: ')
x=len(s)
a=s[-1]+s[1:x-1]+s[0]
print(a)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q31.py
Enter a string: Hello
olleH
```

Q15) Write a Python program to multiply all the elements in a list.

CODE

```
l=eval(input('Enter a list: '))
p=1
for i in l:
    p*=i
print(p)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q32.py
Enter a list: (1,2,3,4,5,6,7,8,9,10)
3628800
```

Q16) Write a Python program to get the smallest number from a list.

CODE

```
l=eval(input('Enter a list: '))
```

```
print(min(l))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q33.py
Enter a list: [1,2,3,0,4,5,6]
0
```

Q17) Write a Python program to append a list to the second list.

CODE

```
l1=eval(input('Enter a list: ')) l2=eval(input('Enter a list: ')) l1.extend(l2) print(l1)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q34.py
Enter a list: [1,2,3,4]
Enter a list: [4,5,6,7,8]
[1, 2, 3, 4, 4, 5, 6, 7, 8]
```

Q18) Write a Python program to generate and print a list of first five and last five elements where the values are square of numbers between one and 30 (both included).

CODE

```
l=[]
```

```
for i in range(1,31):
```

```
    l.append(i**2)
```

```
x=l[:5]+l[-5:]
```

```
print(x)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q35.py
[1, 4, 9, 16, 25, 676, 729, 784, 841, 900]
```

Q19) Write a Python program to get unique values from a list.

CODE

```
l=eval(input('Enter a List: '))
```

```
for i in l:
```

```
    if l.count(i)==1:
```

```
        print(i)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q36.py
Enter a List: (1,2,3,1,4,5,3,6,9,8,7,4)
2
5
6
9
8
7
```

Q20) Write a python program to convert a string to a list.

CODE

```
s=input('Enter a string: ')
```

```
l=[]
```

```
l.append(s)
```

```
print(l)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q37.py
Enter a string: This is a Test
(['This is a Test'])
```

Q21) Write a Python script to concatenate the following dictionaries to create a new one:

```
d1: {'A':1, 'B':2, 'C': 3}
```

```
d2: {'D':4}
```

Output should be:

```
{'A':1, 'B':2, 'C':3, 'D':4}
```

CODE

```
d1=eval(input("Enter a Dictionary: "))
d2=eval(input("Enter a Dictionary: "))
d1.update(d2) print(d1)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q38.py
Enter a Dictionary: {'A':1, 'B':2, 'C': 3}
Enter a Dictionary: {'D':4}
{'A': 1, 'B': 2, 'C': 3, 'D': 4}
```

Q22) Write a Python script to check if a given key already exists in a dictionary.

CODE

```
d=eval(input('Enter a dictionary: '))
x=input('Enter a key: ')
if x in d:
    print('Exists')
else:
    print("Key Does Not Exist")
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q39.py
Enter a dictionary: {'A':1, 'B':2, 'C':3, 'D':4}
Enter a key: B
Exists

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q39.py
Enter a dictionary: {'A':1, 'B':2, 'C':3, 'D':4}
Enter a key: E
Key Does Not Exist
```

Q23) Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Sample Dictionary:

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121,  
12: 144, 13: 169, 14: 196, 15: 225}
```

CODE

```
d={}
for i in range(1,16):
    d[i]=i**2
print(d)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q40.py
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
```

Q24) Write a Python program to sort a dictionary by key.

CODE

```
d=eval(input('Enter a dictionary: '))
print(dict(sorted(d.items())))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_1_Review_of_Python_Basics_Back_Exercise/Q42.py
Enter a dictionary: { 'c': 400,'a': 300, 'b': 200}
{'a': 300, 'b': 200, 'c': 400}
```

Chapter 2

Functions

Q1) Write a function, calculate_area(), that takes base and height as input arguments and returns the area of a triangle as an output. The formula used is:

Triangle Area=1/2*base*height

CODE

```
def calculate_area(B,H):
```

$$A=(1/2)*B*H$$

```
    return A
```

```
b=float(input("Enter the Base of Triangle: "))
```

```
h=float(input("Enter the Height of Triangle: "))
```

```
print("The Area of the given Triangle is",calculate_area(b,h))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q2.py
Enter the Base of Triangle: 12
Enter the Height of Triangle: 12
The Area of the given Triangle is 72.0
```

Q2) Modify the function given in the previous question to take a third parameter called shape type. Shaped type should be either triangle or rectangle. Based on the shape, it should calculate the area.

Formula used: Rectangle Area=length*height

CODE

```
def calculate_area(Shape_Type,B,H):
```

```
    if Shape_Type=="Rectangle":
```

$$A=B*H$$

```

elif Shape_Type=="Triangle":
    A=(1/2)*B*H
    return A

shape=input("Enter the Type of Shape: ")
if shape.lower() in ("rect","rectangle","r","sqaure","s"):
    shape="Rectangle"
elif shape.lower() in ("triangle","t","tri"):
    shape="Triangle"
else:
    print("Shape NOT DEFINED")
    quit()

```

b=float(input("Enter the Dimensions: ")) h=float(input("Enter the Dimensions: ")) print("The Area of the given Figure is",calculate_area(shape,b,h))

OUTPUT

```

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q3.py
Enter the Type of Shape: rect
Enter the Dimensions: 15
Enter the Dimensions: 15
The Area of the given Figure is 225.0

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q3.py
Enter the Type of Shape: tri
Enter the Dimensions: 15
Enter the Dimensions: 12
The Area of the given Figure is 90.0

```

Q3) Write a function, print_pattern(), that takes integer number as argument and print the following pattern if the input is 3.

*
**

If the input is 4, then it should print:

*
**

CODE

```
def pattern(N):  
  
    for i in range(1,N+1):  
        for j in range(i):  
            print("*",end=" ")  
        print()  
  
n=int(input("Enter a Number: "))  
pattern(n)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q4.py  
Enter a Number: 3  
*  
**  
***  
  
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q4.py  
Enter a Number: 4  
*  
**  
***  
****
```

**Q4) Write a function that takes amount dollars and dollar-to-rupee conversion price and then returns the amount converted to rupees.
Create the function in both void and non-void forms.**

CODE

```
def Void(M):  
  
    print(M*85.68)
```

```
def Non_Void(M):
    x=M*85.68
    return int(x)

m=float(input("Enter the Money in USD: "))
Void(m)
print(Non_Void(m))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q17.py
Enter the Money in USD: 5
428.40000000000003
428
```

Q5) Write a function to calculate the value of a box with appropriate default values for its parameters. Your function should have the following input parameters:

- i. Length of box.
- ii. Width of box.
- iii. Height of box.

Tested by writing a complete program to invoke it.

CODE

```
def Volume(L,W,H):
    V=L*W*H
    return V

l=float(input("Enter the Length of Box: "))
w=float(input("Enter the Width of Box: "))
h=float(input("Enter the Height of Box: "))
print("The Volume of the given Box is",Volume(l,w,h))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q18.py
Enter the Length of Box: 8
Enter the Width of Box: 9
Enter the Height of Box: 10
The Volume of the given Box is 720.0
```

Q6) Write a program to find the greatest common divisor between two numbers.

CODE

```
def GCF(N1,N2):
```

```
    l1=[] l2=[] for i in
range(1,N1+1):
```

```
        if N1%i==0:
```

```
            l1.append(i)
```

```
    for i in l1:
```

```
        if N2%i==0:
```

```
            l2.append(i)
```

```
    print("The Greatest Common Integer is",max(l2))
```

```
n1=int(input("Enter a Number: "))
```

```
n2=int(input("Enter a Number: "))
```

```
GCF(n1,n2)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q19.py
Enter a Number: 100
Enter a Number: 90
The Greatest Common Integer is 10
```

Q7) Write a Python function to multiply all the numbers in a list.

Sample List: (8,2,3,-1,7)

Expected Output: -336

CODE

```
def Multiply(l1):
    p=1
    for i in l1:
        p*=i
    return p

l=eval(input("Enter a Number List: "))
print("The Product of Elements of given list is",Multiply(l))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q20.py
Enter a Number List: (8,2,3,-1,7)
The Product of Elements of given list is -336
```

Q8) Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number whose factorial is to be calculated as the argument.

CODE

```
def Multiply(l1):
    p=1
    for i in range(1,l1+1):
        p*=i
    return p

l=eval(input("Enter a Number List: "))
print("The Factorial of",l,"is",Multiply(l))
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q21.py
Enter a Number List: 10
The Factorial of 10 is 3628800
```

Q9) Write a Python function that takes a number as a parameter and checks whether the number is prime or not.

CODE

```
def Prime(l1):
```

```

a=0
for i in range(2,l1):
    if l1%i==0:
        a=0
        break
    else:
        a=1
if a==1:
    print("Number is Prime")
else:
    print("Number is NOT Prime")
l=eval(input("Enter a Number List: "))

```

Prime(l)

OUTPUT

```

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q22.py
Enter a Number List: 19
Number is Prime

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q22.py
Enter a Number List: 20
Number is NOT Prime

```

Q10) Write a Python function that checks whether a passed string is a palindrome or not.

CODE

```

def palindrome(x):
    i=""
    for j in x:
        if j.isalpha()==True:
            i+=j
    if i==i[::-1]:
        print("It is a Palindrome")

```

```
else:  
    print("It is NOT a Palindrome")
```

s=input("Enter a Input: ")

palindrome(s)

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q23.py  
Enter a Input: racecar  
It is a Palindrome
```

Q11) Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated after sorting them alphabetically.

Sample Items: green-red-yellow-black-white

Expected Result: black-green-red-white-yellow-

CODE

```
def Sorter(s):
```

```
    l1=s.split("-")
```

```
    l1.sort()
```

```
    for i in l1:
```

```
        print(i,end="-")
```

S=input("Enter a input: ")

Sorter(S)

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q24.py  
Enter a input: green-red-yellow-black-white  
black-green-red-white-yellow-
```

Q12) Write a method in Python to find and display prime numbers from 2 to N. The value of N should be passed as an argument to the method.

CODE

```
def P_S(N):
    for i in range(2,N+1):
        a=0
        for j in range(2,i):
            if j%i==0:
                a=0
                break
            else:
                a=1
                break
        if a==1:
            print(i,: Number is Prime")
        else:
            print(i,: Number is NOT Prime")
```

```
I=eval(input("Enter a Number List: "))
```

```
P_S(I)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_2_Functions/Q27.py
```

```
Enter a Number List: 20
```

```
2 : Number is NOT Prime
3 : Number is Prime
4 : Number is Prime
5 : Number is Prime
6 : Number is Prime
7 : Number is Prime
8 : Number is Prime
9 : Number is Prime
10 : Number is Prime
11 : Number is Prime
12 : Number is Prime
13 : Number is Prime
14 : Number is Prime
15 : Number is Prime
16 : Number is Prime
17 : Number is Prime
18 : Number is Prime
19 : Number is Prime
20 : Number is Prime
```

Chapter 4

Data File Handling

Q1) File ‘sports.dat’ contains information in the following format:

EventName, Participant.

Write a function that read contents from file ‘sports.dat’ and create a file named ‘Athletic.dat’, copying only those records from ‘sports.dat’ in which the event name is ‘Athletics’.

CODE

```
import pickle  
ch = input("Make a New Sports file? (yes/no): ")  
y = ("yes", "y")  
n = ("no", "n")  
if ch.lower() in y:  
  
    with open("sports.dat", "wb") as f:  
        while True:  
            l = eval(input("Enter Data (e.g., ('Type', 'Sport', 'Participant')): "))  
            pickle.dump(l, f)  
            c = input("Continue? (yes/no): ")  
            if c.lower() in n:  
                break  
  
def Athletics():  
    with open("sports.dat", "rb") as f:  
        with open("Athletics.dat", "wb") as f2:  
            try:  
                while True:  
                    x = pickle.load(f)  
                    if isinstance(x, (list, tuple)) and len(x) > 0 and x[0].lower() == "athletics":
```

```

    pickle.dump(x, f2)

except EOFError:
    print("File Loading Completed!")

with open("Athletics.dat", "rb") as f:
    try:
        while True:
            print(pickle.load(f))
    except EOFError:
        print("End of File!!")

```

Athletics()

OUTPUT

```

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q15.py
Make a New Sports file? (yes/no): yes
Enter Data (e.g., ('Type','Sport','Participant')): ('athletics','Badminton','A')
Continue? (yes/no): y
Enter Data (e.g., ('Type','Sport','Participant')): ('Motorsports','Formula 1','MV')
Continue? (yes/no): n
File Loading Completed!
('athletics', 'Badminton', 'A')
End of File!!

```

Q2) Write a program to count the words 'to' and 'the' present in text file 'Poem.txt'.

CODE

```
f=open("Poem.txt")
```

```
lines=f.read()
```

```
f.close()
```

```
count=0
```

```
line=lines.split()
```

```
x=line.index("to")
```

```
y=line.index("the")
```

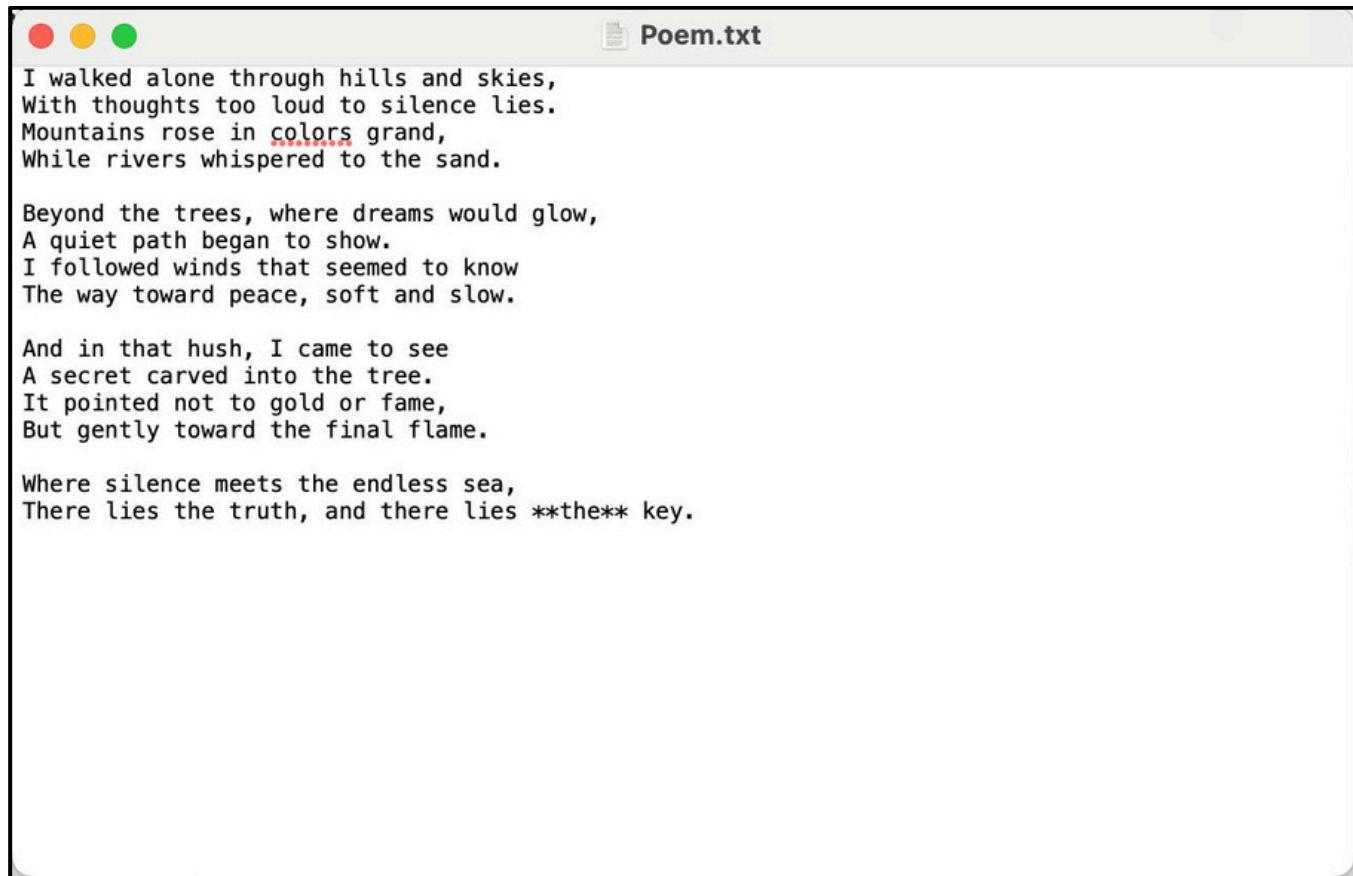
```
for i in range(x,y+1):
```

```
    count+=1
```

```
print(count)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q16.py
13
```



Q3) Write a program to count the number of uppercase alphabets present in text file 'Poem.txt'.

CODE

```
f=open("Poem.txt")
x=f.read()
f.close()
count=0
y="QWERTYUIOPASDFGHJKLZXCVBNM"
for i in x:
    if i in y:
        count+=1
print(count)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q17.py
```

```
15
```

The screenshot shows a text editor window with a light gray background. At the top, there are three colored window control buttons (red, yellow, green) on the left and a title bar on the right with the text "Poem.txt". The main area contains the following poem:

```
I walked alone through hills and skies,  
With thoughts too loud to silence lies.  
Mountains rose in colors grand,  
While rivers whispered to the sand.  
  
Beyond the trees, where dreams would glow,  
A quiet path began to show.  
I followed winds that seemed to know  
The way toward peace, soft and slow.  
  
And in that hush, I came to see  
A secret carved into the tree.  
It pointed not to gold or fame,  
But gently toward the final flame.  
  
Where silence meets the endless sea,  
There lies the truth, and there lies **the** key.
```

Q4) Write a program that copies one file to another and reads the filenames from the user.

CODE

```
b=input("Enter a File Name/Path to read from: ")
```

```
with open(b,"r") as f:
```

```
    MyS=f.read()
```

```
    f.close()
```

```
a=input("Enter a New Name for Location on Desktop: ")
```

```
with open(a,"w") as F:
```

```
    F.writelines(MyS)
```

```
    F.close()
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q18.py
```

```
Enter a File Name/Path to read from: Poem.txt
```

```
Enter a New Name for Location on Desktop: New_Poem.txt
```



Poem.txt

```
I walked alone through hills and skies,  
With thoughts too loud to silence lies.  
Mountains rose in colors grand,  
While rivers whispered to the sand.  
  
Beyond the trees, where dreams would glow,  
A quiet path began to show.  
I followed winds that seemed to know  
The way toward peace, soft and slow.  
  
And in that hush, I came to see  
A secret carved into the tree.  
It pointed not to gold or fame,  
But gently toward the final flame.  
  
Where silence meets the endless sea,  
There lies the truth, and there lies **the** key.
```



New_Poem.txt

```
I walked alone through hills and skies,  
With thoughts too loud to silence lies.  
Mountains rose in colors grand,  
While rivers whispered to the sand.  
  
Beyond the trees, where dreams would glow,  
A quiet path began to show.  
I followed winds that seemed to know  
The way toward peace, soft and slow.  
  
And in that hush, I came to see  
A secret carved into the tree.  
It pointed not to gold or fame,  
But gently toward the final flame.  
  
Where silence meets the endless sea,  
There lies the truth, and there lies **the** key.
```

Q5) Write a program that appends the contents of one file to another and takes the filenames from the user.

CODE

```
b=input("Enter a File Name/Path to read from: ")
```

with open(b,"r") as f:

```
    MyS=f.read()
```

```
    f.close()
```

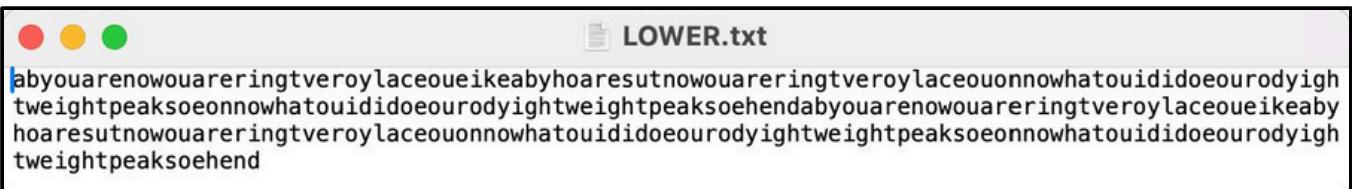
```
a=input("Enter a New Name for Location on Desktop: ")
```

with open(a,"a") as F:

```
    F.writelines(MyS)
```

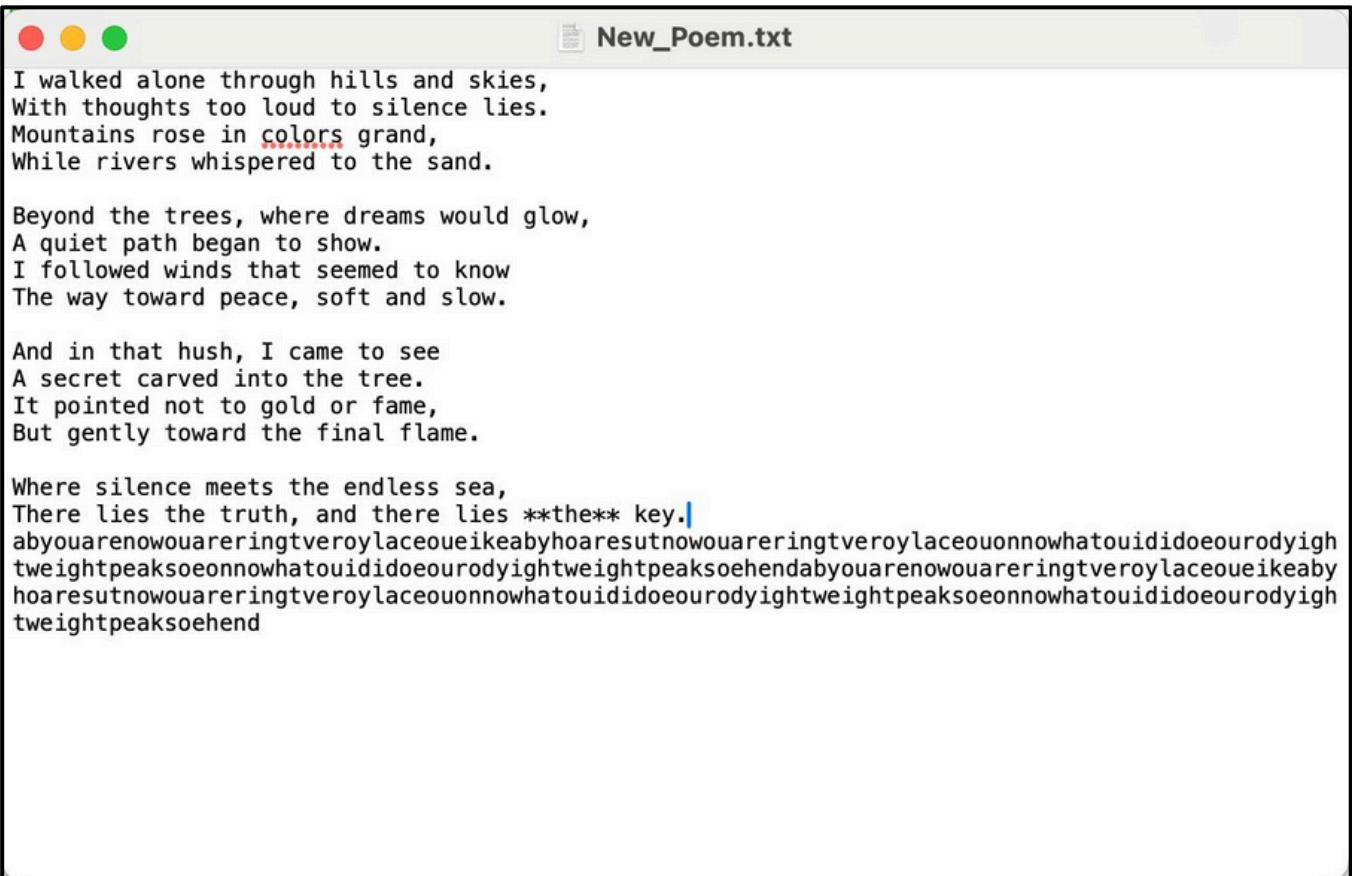
OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q19.py
Enter a File Name/Path to read from: Lower.txt
Enter a New Name for Location on Desktop: New_Poem.txt
```



The screenshot shows a text editor window titled "LOWER.txt". The content of the file is a poem that has been converted to uppercase. The诗的内容是：

```
ab you are now ou ar er ing tver oyl ace ou e ike ab yho ares ut now ou ar er ing tver oyl ace ou on no what ou id id oe ou rody igh  
tweight peak soe on no what ou id id oe ou rody ight weight peak soe h end ab you are now ou ar er ing tver oyl ace ou e ike ab y  
ho ares ut now ou ar er ing tver oyl ace ou on no what ou id id oe ou rody ight weight peak soe on no what ou id id oe ou rody igh  
tweight peak soe h end
```



The screenshot shows a text editor window titled "New_Poem.txt". The content of the file is a poem in its original lowercase form. The诗的内容是：

```
I walked alone through hills and skies,  
With thoughts too loud to silence lies.  
Mountains rose in colors grand,  
While rivers whispered to the sand.  
  
Beyond the trees, where dreams would glow,  
A quiet path began to show.  
I followed winds that seemed to know  
The way toward peace, soft and slow.  
  
And in that hush, I came to see  
A secret carved into the tree.  
It pointed not to gold or fame,  
But gently toward the final flame.  
  
Where silence meets the endless sea,  
There lies the truth, and there lies **the** key.
```

Q6) Write a program that reads characters from the keyboard one by one. All lowercase characters get stored in the file 'LOWER', all

uppercase characters get stored in the file 'UPPER' and all the other characters get stored in the file 'OTHERS'.

CODE

```
f=open("Poem.txt")
a=f.read()
f.close()
l="asdfghjklqwertyuiopzxcvbnm"
u="QWERTYUIOPASDFGHJKLZXCVBNM"
for i in a:
    if i in l:
        with open("LOWER.txt","a") as f:
            f.write(i)
    elif i in u:
        with open("UPPER.txt","a") as f:
            f.write(i)
    elif i not in l and i not in u:
        with open("OTHER.txt","a") as f:
            f.write(i)
```

OUTPUT

```
IWMWBAITAIAIBWT
```

```
walkedalonethroughhillsandskiesiththoughtstooloudtosilenceceliesountainsroseincolorsgrandhileriver
swhisperedtothesandeyondthetreeswheredreamswouldglowquietpathbegantoshownfollowedwindsthatseemedt
oknowhewaytowardpeacesoftandslowdinthathushcametoseesecretcarvedintothreetpointednottogoldorf
ameutgentlytowardthefinalflameheresilencemeetsheendlessseahereliesttheruthandthereliestthekey
```

```
'  
'  
'  
'  
'  
'  
'  
'  
'  
'  
','****.
```

I walked alone through hills and skies,
With thoughts too loud to silence lies.
Mountains rose in colors grand,
While rivers whispered to the sand.

Beyond the trees, where dreams would glow,
A quiet path began to show.
I followed winds that seemed to know
The way toward peace, soft and slow.

And in that hush, I came to see
A secret carved into the tree.
It pointed not to gold or fame,
But gently toward the final flame.

Where silence meets the endless sea,
There lies the truth, and there lies **the** key.

Q7) Consider binary file 'items.dat' containing records stored in the given format:

{item_id: [item_name, amount]}

Write a function, `copy_new()`, that copies all records whose amount is greater than 1000 from 'items.dat' to 'new_items.dat'.

CODE

```
import pickle
```

```
with open("items.dat","wb") as f:
```

```

d={}
c=input("Press Enter to Continue")
while c.lower()!='n':
    x=input("Enter      Item      ID:      ")
    y=input("Enter      Item      Name:      ")
    z=int(input("Enter Amount of Item: "))
    l=[]  l.append(y)  l.append(z)  d[x]=l
    c=input("Continue? ")

```

pickle.dump(d,f)

def copy_new():

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

try:

d=pickle.load(f)

except:

print("End of File!!")

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

for i in d:

if d[i][1]>=1000:

pickle.dump(d[i],f)

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

try:

while True:

print(pickle.load(f))

except:

print("End of File!!")

copy_new()

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q38.py
Press Enter to Continue
Enter Item ID: 1
Enter Item Name: A
Enter Amount of Item: 2499
Continue?
Enter Item ID: 2
Enter Item Name: B
Enter Amount of Item: 1199
Continue?
Enter Item ID: 3
Enter Item Name: C
Enter Amount of Item: 499
Continue?
Enter Item ID: D
Enter Item Name: D
Enter Amount of Item: 2999
Continue? n
[('A', 2499)
 ('B', 1199)
 ('D', 2999)
End of File!!
```

Q8) Anant has been asked to display the names of all students who have scored less than 40 for Remedial Classes.

Write a user-defined function to display the names of the students from the binary file 'Student.dat' who have less than 40.

CODE

```
import pickle
with open("Students.dat","wb") as f:
    d={}
    c=input("Press Enter to Continue")
    while c.lower()!='n':
        x=input("Enter Student Name: ")
        l=int(input("Enter Student Marks: "))
        d[x]=l
        c=input("Continue? ")
    pickle.dump(d,f)
```

```
def copy_new():
    with open("Students.dat","rb") as f:
        try:
            d=pickle.load(f)
        except:
            print("End of File!!")
    with open("Remedial.dat","wb") as f:
        for i in d:
            if d[i]<=40:
                pickle.dump(i,f)
    with open("Remedial.dat","rb") as f:
        try:
            while True:
                print(pickle.load(f))
        except:
            print("End of File!!")
copy_new()
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q43.py
Press Enter to Continue
Enter Student Name: A
Enter Student Marks: 87
Continue?
Enter Student Name: B
Enter Student Marks: 39
Continue?
Enter Student Name: C
Enter Student Marks: 25
Continue?
Enter Student Name: D
Enter Student Marks: 65
Continue? n
B
C
End of File!!
```

Q9) Given a binary file, 'STUDENT.dat', containing records of the following type:

[S_Admno, S_Name, Percentage]

Where these three values are:

S_Admno - Admission Number of student (string)

S_Name - Name of student (string)

Percentage - percentage obtained by student (float)

Write a function in Python that would read the contents of the file 'STUDENT.dat' and that would display the details of those students whose percentage is below 65.

CODE

```
import pickle
```

```
with open("items.dat","wb") as f:
```

```
    d={}
```

```
    c=input("Press Enter to Continue")
```

```
    while c.lower()!='n':
```

```
        x=input("Enter Student Admission No.: ")
```

```
        y=input("Enter      Student      Name:      ")
```

```
        z=int(input("Enter      Percentage:      "))  l=[]
```

```
        l.append(y)      l.append(z)      d[x]=l
```

```
        c=input("Continue? ")
```

```
    pickle.dump(d,f)
```

```
def copy_new():
```

```
    with open("items.dat","rb") as f:
```

```
        try:
```

```
            d=pickle.load(f)
```

```
        except:
```

```

print("End of File!!")

with open("new_items.dat","wb") as f:
    for i in d:
        if d[i][1]<=65:
            D={}
            D[i]=d[i]
            pickle.dump(D,f)

with open("new_items.dat","rb") as f:
    try:
        while True:
            print(pickle.load(f))
    except:
        print("End of File!!")

```

copy_new()

OUTPUT

```

= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q45.py
Press Enter to Continue
Enter Student Admission No.: 1
Enter Student Name: A
Enter Percentage: 87
Continue?
Enter Student Admission No.: 2
Enter Student Name: B
Enter Percentage: 40
Continue?
Enter Student Admission No.: 3
Enter Student Name: C
Enter Percentage: 64
Continue?
Enter Student Admission No.: 4
Enter Student Name: D
Enter Percentage: 70
Continue? n
{'2': ('B', 40)}
{'3': ('C', 64)}
End of File!!

```

Q10) Create CSV file 'Groceries' to store information of different items existing in a shop. The information is to be stored with respect to each item code, name, price, qty. Write a program to accept the data from the user and store it permanently in the CSV file.

CODE

```
import csv

try:

    with open("Groceries.csv", "r") as f_check:
        is_empty = f_check.readline() == ""

    except FileNotFoundError:
        is_empty = True

    with open("Groceries.csv", "a", newline="") as f:
        writer = csv.writer(f)
        if is_empty:
            writer.writerow(["Item Code", "Item Name", "Price", "Quantity"])
        num_rows = int(input("Enter number of items to add: "))
        rows = []
        for _ in range(num_rows):
            code = input("Enter Item Code: ")
            name = input("Enter Item Name: ")
            price = input("Enter Price: ")
            qty = input("Enter Quantity: ")
            rows.append([code, name, price, qty])
        writer.writerows(rows)

    with open("Groceries.csv", "r", newline="") as f:
        reader = csv.reader(f)
        next(reader)
        for row in reader:
            print(row)
```

OUTPUT

```
= RESTART: /Users/praty/Documents/CS Practicals 2025-26/Chapter_4_Data_File_Handling/Back Exercise/Q52.py
Enter number of items to add: 2
Enter Item Code: 1
Enter Item Name: A
Enter Price: 9.99
Enter Quantity: 12
Enter Item Code: 2
Enter Item Name: B
Enter Price: 4.99
Enter Quantity: 24
['1', 'A', '9.99', '12']
['2', 'B', '4.99', '24']
```