**PRESIDENCY SCHOOL**

**BANGALORE SOUTH**



**PROGRAMMING**

**IN**

**PYTHON & SQL**

**Subject: COMPUTER SCIENCE**

**DONE BY:**

Hamza Shabbir Sahapurwala

XII ‘C’

2023-2024

CERTIFICATE

Name: Class: 12th ‘C’

Exam No: \_\_\_\_

This is certified to be the bonafide work of the student in the computer science laboratory during the academic year 2023-2024.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TEACHER INCHARGE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

EXAMINER’S SIGNATURE PRINCIPAL

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Institution Rubber stamp

**INDEX**

|  |  |
| --- | --- |
| **S.NO.** | **PROGRAM** |
| 1] | Write a menu driven program using functions, to find:   1. **Factorial of a number**           The factorial of a number is the product of all the integers from 1 to n.  **Example,**  The factorial of 6 is 1\* 2\* 3\* 4\* 5\* 6 = 720.  Factorial is not defined for negative numbers, and the factorial of zero is one, 0! is 1.   1. **Fibonacci series up to nth term**   It’s a unique sequence where the next number is the sum of the previous two numbers. Where the first two terms are always 0 and 1.  The series Looks like : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …   1. **Check if a given number is Palindrome or not.**   A palindrome is nothing but any number or a string which remains unaltered when reversed.  **Example:**12321 **Output:**Yes, a Palindrome number |
| 2] | Write a menu driven program using functions, to find:   1. **Check a given number is Armstrong number or not**   A positive integer is called an Armstrong number of order n if  **abcd….. = an + bn + cn + dn …………**  In case of an Armstrong number of 3 digits, the sum of cubes of each digit is equal to the number itself.  **Example:** 153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3  153 is an Armstrong number.   1. **To display Floyd's triangle**   Eg:  **Enter the number of Rows: 5**  **Floyd's Triangle:**  **1**  **2        3**  **4        5         6**  **7        8         9         10**  **11 12       13       14       15** |
| 3] | Write a menu driven program for the following:   1. Define a function **MSEARCH(STATES)** to display all the names from the list STATES, which are starting with the alphabet M.   For eg. If list STATES contains  **L = ["MP", "UP", "DL", "UK", "KA", “MH”]**  The output should be:  **[“MP”, “MH”]**   1. A function with the following signature:   **remove\_letter(sentence, letter)**        This function should take a string & a letter as arguments, returning a copy of that string with        every instance of the indicated letter removed.        For eg:  **remove\_letter("Hello there!", "e")**        Should return the string **"Hllo thr!"**   1. Enter a string & count number of words in the given string.   **Suppose the string is:**  **“I am fine**”  Output should be:  **Number of words = 3** |
| 4] | Write a menu driven program in Python using user defined functions that take a list as parameter and return (Do not use built-in functions):   1. Maximum of the element 2. Minimum of the element 3. Sum of the elements. |
| 5] | Write a menu driven program in Python using user defined functions for the following:   1. To read and display the text file content line by line with each word separated by “$”. 2. To remove all the lines that contain the character ‘a’ in a file and write it to another file. |
| 6] | Write a menu driven program with the following functions:   1. **CreateTextFile( )** - create a text file "data.txt" with several lines of text 2. **CopyVowelWord( )** - create another text file "vowel.txt" which will store all the words starting with vowel from "data.txt". 3. Read & display the contents of both the files. 4. Display the total number of words starting with vowel. |
| 7] | Write a menu driven program with the following functions:   1. **CreateTextFile( )** - create a text file **" content.txt"** with few lines. 2. **CountAll( )** - count number of lines, consonants, digits, spaces & words. 3. **ReplaceSpace( )** - create another file called **"wspace.txt"** using the original which will contain the text after replacing all the blank spaces with '#'. 4. Read & display the contents of both the files. |
| 8] | Write a menu driven program with the following functions:   1. **DISPLAYWORDS( )** to read lines from text file **STORY.TXT** , and display those words which are less than 4 characters. 2. **SEARCHWORD( )** to search a word and its frequency in a text file. |
| 9] | Write a menu driven program in Python using Pickle library and   1. Create a binary file **“STUDENT.DAT”** with following structure:  * Admission\_number * Student Name * Percentage  1. Write a function **countrec( )** in Python that would read contents of the file **“STUDENT.DAT”** and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%. |
| 10] | Write a menu driven program in Python using **Pickle library** and   1. Create a binary file with following structure:  * CompanyID * Company name * Turnover  1. Display the contents of the binary file. 2. Display the Company whose turnover is above user given value. 3. Search a Company by Company ID given by user. |
| 11] | Write a menu driven program in Python using **Pickle library** and   1. Create binary file with following structure:  * Travelid * From * To  1. Append data to the file. 2. Update a record based on travelid. 3. Display the contents of the binary file. |
| 12] | Write a menu driven program in Python:   1. Define a function to write the following data into a **CSV file**:  * Roll no * Name of student * Mark in Sub1 * Mark in sub2 * Mark in sub3 * Mark in sub4 * Mark in sub5   **Perform following operations on the CSV file:**   1. Define a function to read the CSV file and calculate total and percentage for each student. 2. Define a function to display the name of student if in any subject marks are greater than 80% (Assume marks are out of 100) |
| 13] | Write a menu driven program in Python:   1. Define a function to write the following data into a CSV file “**emp.CSV**”  * empid * empName * Salary * Department  1. Define a function to search the record from “**emp.CSV**” file based on empid. If record found display the record, otherwise display appropriate message. |
| 14] | A list contains the following record of a Hostel:  **[Hostel\_No, Total\_Students, Total Rooms]**  Write a menu driven program, with the following user defined functions to perform given operations on the stack named **‘Hostel’**:   1. **Push\_element( )** - To push an object containing Hostel\_No and Total  Students along with Total Rooms to the stack 2. **Pop\_element( )** - To pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.   **For example:**    If the lists of Hostel details are:  **[1, 2000, 1000]**  **[2, 1500, 800]**  **[3, 5000, 2000]**   The output should be:  **[3, 5000, 2000]**  **[2, 1500, 800]**  **[1, 2000, 1000]**  **Stack Empty**   1. Implement all the stack operations. |
| 15] | Mr. John wants to store book details like book ID and Book Name in a dictionary format. Write a menu driven program, with separate user defined functions to perform the following operations:   1. Push the values (Name of the book) of the dictionary into a stack where book names begin with A or C. 2. Pop and display the content of the stack.   For example, if the content of Dictionary is as follows:  **Book={"B001":"C++", "B002":"Python", "B003":"Ada", "B004":"C",**  **"B005":"Java", "B006":"Oracle", "B007":"HTMl"}**  The output of the program should be : **C   Ada   C++**   1. Implement all the stack operations. |

PYTHON

PROGRAMS

**Program 1:**

Write a menu driven program using functions, to find:

1. **Factorial of a number**

        The factorial of a number is the product of all the integers from 1 to n.

**Example,**

The factorial of 6 is 1\* 2\* 3\* 4\* 5\* 6 = 720.

Factorial is not defined for negative numbers, and the factorial of zero is one, 0! is 1.

1. **Fibonacci series up to nth term**

It’s a unique sequence where the next number is the sum of the previous two numbers.  
Where the first two terms are always 0 and 1.

The series Looks like : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …

1. **Check if a given number is Palindrome or not.**

A palindrome is nothing but any number or a string which remains unaltered when reversed.

**Example:**12321  
**Output:**Yes, a Palindrome number

**Code:**

def factorial():

"""Calculating Factorial of a Number"""

a = int(input('Enter a no.:\n'))

b = 1

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

b \*= i

print(f'The Factorial of the no. is {b}.')

def fib():

""" Function to generate Fibonacci series"""

a = int(input('Enter a no.:\n'))

e, b, c = 0, 1, 0

print(e, b, end=' ')

for i in range(a-2):

c = e+b

print(c, end=' ')

e = b

b = c

print()

def palin():

""" Palindrome checker"""

a = int(input('Enter a no.:\n'))

b = 10 \*\* (len(str(a))-1)

c = a

e = 0

while a != 0:

d = (a % 10) \* b

e += d

a //= 10

b //= 10

if e == c:

print('Its a Palindrome.')

return

print('Its not a Palindrome.')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to find Factorial of a number.

Enter 2 to generate Fibonacci Series for x no.

Enter 3 to check if a number is palindrome or not.\n'''))

if n == 1:

factorial()

elif n == 2:

fib()

elif n == 3:

palin()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to find Factorial of a number.

Enter 2 to generate Fibonacci Series for x no.

Enter 3 to check if a number is palindrome or not.

1

Enter a no.:

5

The Factorial of the no. is 120.

Do you want to try again? (Y/N)

y

Enter 1 to find Factorial of a number.

Enter 2 to generate Fibonacci Series for x no.

Enter 3 to check if a number is palindrome or not.

2

Enter a no.:

8

0 1 1 2 3 5 8 13

Do you want to try again? (Y/N)

y

Enter 1 to find Factorial of a number.

Enter 2 to generate Fibonacci Series for x no.

Enter 3 to check if a number is palindrome or not.

3

Enter a no.:

151

Its a Palindrome.

Do you want to try again? (Y/N)

n

**Program 2:**

Write a menu driven program using functions, to find:

1. **Check a given number is Armstrong number or not**

A positive integer is called an Armstrong number of order n if

**abcd….. = an + bn + cn + dn …………**

In case of an Armstrong number of 3 digits, the sum of cubes of each digit is equal to the number itself.

**Example:** 153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3

153 is an Armstrong number.

1. **To display Floyd's triangle**

Eg:

**Enter the number of Rows: 5**

**Floyd's Triangle:**

**1**

**2        3**

**4        5         6**

**7        8         9         10**

**11 12       13       14       15**

**Code:**

def ams():

""" Armstrong checker"""

a = int(input('Enter a no.:\n'))

b = len(str(a))

c = a

e = 0

while a != 0:

d = (a % 10) \*\* b

e += d

a //= 10

if c == e:

print('It is an Armstrong no.')

return

print('Its not an Armstrong no.')

def floyd():

""" Function to generate Floyd's Triangle"""

a = int(input('Enter the number of Rows:\n'))

c = 1

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

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

print(c, end=' ')

c+=1

print()

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to find if a No. is Amstrong or not.

Enter 2 to generate Floyd's Triangle for x no.\n'''))

if n == 1:

ams()

elif n == 2:

floyd()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to find if a No. is Amstrong or not.

Enter 2 to generate Floyd's Triangle for x no.

1

Enter a no.:

153

It is an Armstrong no.

Do you want to try again? (Y/N)

y

Enter 1 to find if a No. is Amstrong or not.

Enter 2 to generate Floyd's Triangle for x no.

2

Enter the number of Rows:

5

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

Do you want to try again? (Y/N)

n

**Program 3:**

Write a menu driven program for the following:

1. Define a function **MSEARCH(STATES)** to display all the names from the list STATES, which are starting with the alphabet M.

For eg. If list STATES contains

**L = ["MP", "UP", "DL", "UK", "KA", “MH”]**

The output should be:

**[“MP”, “MH”]**

1. A function with the following signature:

**remove\_letter(sentence, letter)**

      This function should take a string & a letter as arguments, returning a copy of that string with

      every instance of the indicated letter removed.

      For eg:

**remove\_letter("Hello there!", "e")**

      Should return the string **"Hllo thr!"**

1. Enter a string & count number of words in the given string.

**Suppose the string is:**

**“I am fine**”

Output should be:

**Number of words = 3**

**Code:**

def MSEARCH(STATES):

"""Sorting States from a list"""

for i in STATES:

if i[0] in 'Mm':

print(i)

def remove\_letter(sentence, letter):

""" Removing a letter from a string"""

c = ''

for i in sentence:

if i != letter:

c += i

print('Here is the sentence without the letter:\n', c)

def cnt(sentence):

""" Count the number of words in a sentence"""

if sentence[0] != ' ':

print('Number of words = ', sentence.count(' ')+1)

return

print('Number of words = ', sentence.count(' '))

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to display all the names from the list STATES, which are starting with alphabet M.

Enter 2 to enter a sentence and remove a letter from it.

Enter 3 to count the no. of words in a sentence.\n'''))

if n == 1:

states = ["MP", "UP", "DL", "UK", "KA", 'MH']

MSEARCH(states)

elif n == 2:

sentence = input('Enter A Sentence:\n')

letter = input('Enter A Letter You Want To Remove:\n')

remove\_letter(sentence, letter)

elif n == 3:

sentence = input('Enter A Sentence:\n')

cnt(sentence)

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to display all the names from the list STATES, which are starting with alphabet M.

Enter 2 to enter a sentence and remove a letter from it.

Enter 3 to count the no. of words in a sentence.

1

MP

MH

Do you want to try again? (Y/N)

y

Enter 1 to display all the names from the list STATES, which are starting with alphabet M.

Enter 2 to enter a sentence and remove a letter from it.

Enter 3 to count the no. of words in a sentence.

2

Enter A Sentence:

Betty Bought Butter

Enter A Letter You Want To Remove:

t

Here is the sentence without the letter:

Bey Bough Buer

Do you want to try again? (Y/N)

y

Enter 1 to display all the names from the list STATES, which are starting with alphabet M.

Enter 2 to enter a sentence and remove a letter from it.

Enter 3 to count the no. of words in a sentence.

3

Enter A Sentence:

B & C are married.

Number of words = 5

Do you want to try again? (Y/N)

n

**Program 4:**

Write a menu driven program in Python using user defined functions that take a list as parameter and return (Do not use built-in functions):

1. Maximum of the element
2. Minimum of the element
3. Sum of the elements.

**Code:**

def max\_list(l):

""" Identify the element with the max len from a list"""

a = l[0]

for b in l:

if b > a:

a = b

print('The Maximun of the elements is ', a, '.')

def min\_list(l):

""" Identify the element with the least len from a list"""

a = l[0]

for b in l:

if b < a:

a = b

print('The Minimum of the elements is ', a, '.')

def sum\_list(l):

""" Calculating the sum of a list"""

a = 0

for b in l:

a += b

print('The Sum of the elements are ', a, '.')

d = 'Yy'

while d in 'Yy':

li = eval(input('Enter a list containing no.:\n'))

n = int(input('''Enter 1 to find the highest no. in the list.

Enter 2 to find the lowest value in the list.

Enter 3 to calculate the sum of all the values in the list.\n'''))

if n == 1:

max\_list(li)

elif n == 2:

min\_list(li)

elif n == 3:

sum\_list(li)

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter a list containing no.:

[1,2,5,3,4]

Enter 1 to find the highest no. in the list.

Enter 2 to find the lowest value in the list.

Enter 3 to calculate the sum of all the values in the list.

1

The Maximun of the elements is 5 .

Do you want to try again? (Y/N)

y

Enter a list containing no.:

[1,2,3,4,5]

Enter 1 to find the highest no. in the list.

Enter 2 to find the lowest value in the list.

Enter 3 to calculate the sum of all the values in the list.

2

The Minimum of the elements is 1 .

Do you want to try again? (Y/N)

y

Enter a list containing no.:

[1,2,3,4,5]

Enter 1 to find the highest no. in the list.

Enter 2 to find the lowest value in the list.

Enter 3 to calculate the sum of all the values in the list.

3

The Sum of the elements are 15 .

Do you want to try again? (Y/N)

n

**Program 5:**

Write a menu driven program in Python using user defined functions for the following:

1. To read and display the text file content line by line with each word separated by “$”.
2. To remove all the lines that contain the character ‘a’ in a file and write it to another file.

**Code:**

def separatebycharacter():

""" Replace all whitespace with an emoji"""

c = ''

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

a = f.read()

for i in a:

if i == ' ':

c += '$'

continue

c += i

print(c)

def remove\_a():

""" Remove all lines containing the letter a"""

c = []

b = []

with open('f.txt') as f:

a = f.readlines()

for i in a:

if 'a' in i:

b.append(i)

continue

c.append(i)

if len(b) != 0:

with open('f.txt', 'w') as e:

e.writelines(c)

with open('k.txt', 'w') as d:

d.writelines(b)

else:

print('The File does not have the character\'a\'')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to seperate the words in a txt file with 😊.

Enter 2 to remove any lines containing the character 'a' from a txt file and adding them to new text file.\n'''))

if n == 1:

separatebycharacter()

elif n == 2:

remove\_a()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to seperate the words in a txt file with 😊.

Enter 2 to remove any lines containing the character 'a' from a txt file and adding them to new text file.

1

Betty😊bought😊butter😊for😊the😊first😊time.

A😊man😊stole😊that😊butter.

Do you want to try again? (Y/N)

y

Enter 1 to seperate the words in a txt file with 😊.

Enter 2 to remove any lines containing the character 'a' from a txt file and adding them to new text file.

2

Do you want to try again? (Y/N)

y

Enter 1 to seperate the words in a txt file with 😊.

Enter 2 to remove any lines containing the character 'a' from a txt file and adding them to new text file.

1

Betty😊bought😊butter😊for😊the😊first😊time.

Do you want to try again? (Y/N)

n

**Program 6:**

Write a menu driven program with the following functions:

1. **CreateTextFile( )** - create a text file "data.txt" with several lines of text
2. **CopyVowelWord( )** - create another text file "vowel.txt" which will store all the words starting with vowel from "data.txt".
3. Read & display the contents of both the files.
4. Display the total number of words starting with vowel.

**Code:**

def CreateTextFile():

"""Creates a text file named data while"""

a = '''This contains text that should be input in the data.txt file.

Hello and How are you?

I am fine, Don't Worry About me.'''

with open('data.txt', 'w') as f:

f.write(a)

def CopyVowelWord():

"""To create text file named vowel which will store all the

words starting with vowel from the text file data"""

vowel = []

with open('data.txt') as f:

a = f.read().split()

for i in a:

if i[0] in 'AEIOUaeiou':

vowel.append(i)

with open('vowel.txt', 'w') as v:

f.writelines(vowel)

def display():

"""Displays the content of both the text files"""

print('The Contents of data.txt is:')

with open('data.txt') as f:

print(f.read())

print('The Content of vowel.txt is:')

with open('vowel.txt') as v:

print(v.read())

def TotalNumberOfWordsStartingWithVowel():

"""Displays the no. of words starting with vowels in the data.txt file"""

vowel = 0

with open('data.txt') as f:

a = f.read().split()

for i in a:

if i[0] in 'AEIOUaeiou':

vowel += 1

print(f'The no. of words starting with vowel in data.txt is {vowel}.')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create a text file named data and add some text in it.

Enter 2 to copy all the words starting with a vowel from data and adding them to a new text file named vowel.

Enter 3 to display the content of both the files.

Enter 4 to count the number of words starting with vowel in data.txt.\n'''))

if n == 1:

CreateTextFile()

elif n == 2:

CopyVowelWord()

elif n == 3:

display()

elif n == 4:

TotalNumberOfWordsStartingWithVowel()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create a text file named data and add some text in it.

Enter 2 to copy all the words starting with a vowel from data and adding them to a new text file named vowel.

Enter 3 to display the content of both the files.

Enter 4 to count the number of words starting with vowel in data.txt.

1

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named data and add some text in it.

Enter 2 to copy all the words starting with a vowel from data and adding them to a new text file named vowel.

Enter 3 to display the content of both the files.

Enter 4 to count the number of words starting with vowel in data.txt.

2

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named data and add some text in it.

Enter 2 to copy all the words starting with a vowel from data and adding them to a new text file named vowel.

Enter 3 to display the content of both the files.

Enter 4 to count the number of words starting with vowel in data.txt.

3

The Contents of data.txt is:

This contains text that should be input in the data.txt file.

Hello and How are you?

I am fine, Don't Worry About me.

The Content of vowel.txt is:

inputinandareIamAbout

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named data and add some text in it.

Enter 2 to copy all the words starting with a vowel from data and adding them to a new text file named vowel.

Enter 3 to display the content of both the files.

Enter 4 to count the number of words starting with vowel in data.txt.

4

The no. of words starting with vowel in data.txt is 7.

Do you want to try again? (Y/N)

n

**Program 7:**

Write a menu driven program with the following functions:

1. **CreateTextFile( )** - create a text file **" content.txt"** with few lines.
2. **CountAll( )** - count number of lines, consonants, digits, spaces & words.
3. **ReplaceSpace( )** - create another file called **"wspace.txt"** using the original which will contain the text after replacing all the blank spaces with '#'.
4. Read & display the contents of both the files.

**Code:**

def CreateTextFile7(): # Added 6 to make the func name unique

""" Function to create a text file with a few lines"""

sampleLines = "Lorem ips3um dolor sit amet, 43c43onsectetur adipiscing 4elit. Vestibulum bi3b4endum mollis5 5viverra.3 Nulla a ipsum vitae nisl dignis3sim suscipit vitae at 4nisl. Curabitur3 finibus, sapien ac dic8tum ves3ibulum, eros magna auctor quam, quis eleifend tortor leo imperdiet ipsum. Praesent eget mo38lestie tellus. 8Mauris varius molestie enim nec euismod. Mauris tincidu3t lacinia pharetra. Quisque vel lacus id83 turpis egestas varius. Et83iam risus ipsum, elementum38 ac porta eu, rhoncus sed risus. D8uis vel nibh id dolor aliq8uam rutrum. Nunc ac pharetra ipsum. Fusce variu83s bibendum augue vitae sodales. Nam in nunc in lorem sagittis imperdiet 38a lacinia quam."

with open("content.txt", "w") as f:

f.write(sampleLines)

def CountAll():

""" Calculate lines, consonants, words, digits and whitespaces from a txt file"""

with open("content.txt", "r") as f:

content = f.read()

lines = sum(1 for line in open('content.txt'))

consonants = 0

for i in content.lower():

# Excluding vowels, whitespaces, periods & commas

vowels = ['a', 'e', 'i', 'o', 'u', ' ', ',', '.']

if i not in vowels:

consonants += 1

digits = 0

for char in content:

if char.isdigit():

digits = digits + 1

whitespaces = 0

for char in content:

if char == " ":

whitespaces += 1

words = 0

for word in content.split():

words += 1

return lines, consonants, digits, whitespaces, words # Returns a tuple

def ReplaceSpace():

""" Replacing all whitespaces with # in a file"""

with open("content.txt", "r") as f1:

output = f1.read().replace(' ', '#')

with open("wspace.txt", "w") as f:

f.write(output)

def display():

print('The Content of content.txt is:')

with open("content.txt") as f1:

print(f1.read())

print('The Content of wspace.txt is:')

with open('wspace.txt') as f:

print(f.read())

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create a text file named content and add some text in it.

Enter 2 to count number of lines, consonants, digits, spaces & words.

Enter 3 to create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.

Enter 4 to Read & display the contents of both the files.\n'''))

if n == 1:

CreateTextFile7()

elif n == 3:

ReplaceSpace()

elif n == 2:

print(

f'The no. of lines, consonants, digits, whitespaces and words are\n{CountAll()}')

elif n == 4:

display()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create a text file named content and add some text in it.

Enter 2 to count number of lines, consonants, digits, spaces & words.

Enter 3 to create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.

Enter 4 to Read & display the contents of both the files.

1

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named content and add some text in it.

Enter 2 to count number of lines, consonants, digits, spaces & words.

Enter 3 to create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.

Enter 4 to Read & display the contents of both the files.

2

96

The no. of lines, consonants, digits, whitespaces and words are

(1, 326, 32, 95, 96)

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named content and add some text in it.

Enter 2 to count number of lines, consonants, digits, spaces & words.

Enter 3 to create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.

Enter 4 to Read & display the contents of both the files.

3

Do you want to try again? (Y/N)

y

Enter 1 to create a text file named content and add some text in it.

Enter 2 to count number of lines, consonants, digits, spaces & words.

Enter 3 to create another file called "wspace.txt" using the original which will contain the text after replacing all the blank spaces with '#'.

Enter 4 to Read & display the contents of both the files.

4

The Content of content.txt is:

Lorem ips3um dolor sit amet, 43c43onsectetur adipiscing 4elit. Vestibulum bi3b4endum mollis5 5viverra.3 Nulla a ipsum vitae nisl dignis3sim suscipit vitae at 4nisl. Curabitur3 finibus, sapien ac dic8tum ves3ibulum, eros magna auctor quam, quis eleifend tortor leo imperdiet ipsum. Praesent eget mo38lestie tellus. 8Mauris varius molestie enim nec euismod. Mauris tincidu3t lacinia pharetra. Quisque vel lacus id83 turpis egestas varius. Et83iam risus ipsum, elementum38 ac porta eu, rhoncus sed risus. D8uis vel nibh id dolor aliq8uam rutrum. Nunc ac pharetra ipsum.

Fusce variu83s bibendum augue vitae sodales. Nam in nunc in lorem sagittis imperdiet 38a lacinia quam.

The Content of wspace.txt is:

Lorem#ips3um#dolor#sit#amet,#43c43onsectetur#adipiscing#4elit.#Vestibulum#bi3b4endum#mollis5#5viverra.3#Nulla#a#ipsum#vitae#nisl#dignis3sim#suscipit#vitae#at#4nisl.#Curabitur3#finibus,#sapien#ac#dic8tum#ves3ibulum,#eros#magna#auctor#quam,#quis#eleifend#tortor#leo#imperdiet#ipsum.#Praesent#eget#mo38lestie#tellus.#8Mauris#varius#molestie#enim#nec#euismod.#Mauris#tincidu3t#lacinia#pharetra.#Quisque#vel#lacus#id83#turpis#egestas#varius.#Et83iam#risus#ipsum,#elementum38#ac#porta#eu,#rhoncus#sed#risus.#D8uis#vel#nibh#id#dolor#aliq8uam#rutrum.#Nunc#ac#pharetra#ipsum.#Fusce#variu83s#bibendum#augue#vitae#sodales.#Nam#in#nunc#in#lorem#sagittis#imperdiet#38a#lacinia#quam.

Do you want to try again? (Y/N)

n

**Program 8:**

Write a menu driven program with the following functions:

1. **DISPLAYWORDS( )** to read lines from text file **STORY.TXT** , and display those words which are less than 4 characters.
2. **SEARCHWORD( )** to search a word and its frequency in a text file.

**Code:**

def displaywords():

'''Displays the words from the text file story which are less than 4 characters'''

with open('story.txt') as f:

a = f.read().split()

for i in a:

if len(i) < 4:

print(i)

def seachword():

'''Searches the no. of times a word is repeated inside the file'''

e = input('Enter a word to be searched:\n')

with open('story.txt') as f:

a = f.read().split()

d = a.count(e)

print(f'The word {e} is {d} times repeating in the file.')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to display words which are less than 4 characters in story.txt.

Enter 2 to search a word and its frequency in story.txt.\n'''))

if n == 1:

displaywords()

elif n == 2:

seachword()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to display words which are less than 4 characters in story.txt.

Enter 2 to search a word and its frequency in story.txt.

1

is

a

man

He

saw

was

a

boy

and

The

man

no

to

he

was

now

to

the

boy

and

The

man

the

boy

was

and

an

but

the

man

did

to

you

the

Do you want to try again? (Y/N)

y

Enter 1 to display words which are less than 4 characters in story.txt.

Enter 2 to search a word and its frequency in story.txt.

2

Enter a word to be searched:

man

The word man is 4 times repeating in the file.

Do you want to try again? (Y/N)

n

**Program 9:**

Write a menu driven program in Python using Pickle library and

1. Create a binary file **“STUDENT.DAT”** with following structure:

* Admission\_number
* Student Name
* Percentage

1. Write a function **countrec( )** in Python that would read contents of the file **“STUDENT.DAT”** and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%.

**Code:**

import pickle as p

c = 0

def createandaddtobinaryfile():

'''Creates a binary file names student which stores admission no., student name and percentage in dictionary format'''

s = {}

global c

c = int(input('How many records do you want to add?\n'))

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

for i in range(c):

s['Admission\_number'] = int(input('Enter Admission number:\n'))

s['Student Name'] = input('Enter Student Name:\n')

s['Percentage'] = int(input('Enter Percentage:\n'))

p.dump(s, f)

def countrec():

'''Counts the no. of students who's percentage is above 75'''

a = 0

s = {}

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

for i in range(c):

s = p.load(f)

if s['Percentage'] > 75:

print(s)

a += 1

print(f'The No. of students scoring above 75% are {a}.')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create a binary file named student and add admission number, student name and percentage of students.

Enter 2 to read contents of the file.\n'''))

if n == 1:

createandaddtobinaryfile()

elif n == 2:

countrec()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create a binary file named student and add admission number, student name and percentage of students.

Enter 2 to read contents of the file.

1

How many records do you want to add?

3

Enter Admission number:

1001

Enter Student Name:

Alok

Enter Percentage:

74

Enter Admission number:

1002

Enter Student Name:

Bercan

Enter Percentage:

78

Enter Admission number:

1003

Enter Student Name:

Calvin

Enter Percentage:

76

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named student and add admission number, student name and percentage of students.

Enter 2 to read contents of the file.

2

{'Admission\_number': 1002, 'Student Name': 'Bercan', 'Percentage': 78}

{'Admission\_number': 1003, 'Student Name': 'Calvin', 'Percentage': 76}

The No. of students scoring above 75% are 2.

Do you want to try again? (Y/N)

n

**Program 10:**

Write a menu driven program in Python using **Pickle library** and

1. Create a binary file with following structure:

* CompanyID
* Company name
* Turnover

1. Display the contents of the binary file.
2. Display the Company whose turnover is above user given value.
3. Search a Company by Company ID given by user.

**Code:**

import pickle as p

c = 0

def createbinary():

'''Creates a binary file named company which stores the id of the companies, their names and turnover respectively'''

s = {}

global c

c = int(input('Enter the number of companies:\n'))

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

for i in range(c):

s['CompanyID'] = int(input('Enter Company ID:\n'))

s['CompanyName'] = input('Enter Company Name:\n')

s['Turnover'] = int(input('Enter Turnover:\n'))

p.dump(s, f)

def display():

'''Displays all the companies dictionaries'''

s = {}

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

for i in range(c):

s = p.load(f)

print(s)

def turnover():

'''Prints the turnover of the companies which are higher than the input turnover'''

s = {}

a = int(input('Enter a turnover:\n'))

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

for i in range(c):

s = p.load(f)

if s['Turnover'] > a:

print(s)

def companyid():

'''Searches for a particular company using its id and prints only its data'''

s = {}

a = int(input('Enter Company ID:\n'))

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

for i in range(c):

s = p.load(f)

if s['CompanyID'] == a:

print(s)

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.\n'''))

if n == 1:

createbinary()

elif n == 2:

display()

elif n == 3:

turnover()

elif n == 4:

companyid()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.

1

Enter the number of companies:

2

Enter Company ID:

1001

Enter Company Name:

Orac

Enter Turnover:

20000

Enter Company ID:

1002

Enter Company Name:

Micro

Enter Turnover:

30000

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.

2

{'CompanyID': 1001, 'CompanyName': 'Orac', 'Turnover': 20000}

{'CompanyID': 1002, 'CompanyName': 'Micro', 'Turnover': 30000}

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.

3

Enter a turnover:

25000

{'CompanyID': 1002, 'CompanyName': 'Micro', 'Turnover': 30000}

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.

4

Enter Company ID:

1003

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named company and add company id, company name and turnover.

Enter 2 to read contents of the file.

Enter 3 to input a turnover value and know which Company's turnover is above your given value.

Enter 4 to search for a Company with a Company ID.

4

Enter Company ID:

1001

{'CompanyID': 1001, 'CompanyName': 'Orac', 'Turnover': 20000}

Do you want to try again? (Y/N)

n

**Program 11:**

Write a menu driven program in Python using **Pickle library** and

1. Create binary file with following structure:

* Travelid
* From
* To

1. Append data to the file.
2. Update a record based on travelid.
3. Display the contents of the binary file.

**Code:**

import pickle as p

a = 0

def bina():

'''Creates a binary file named travel which stores the travel id, from and to address respectively'''

s = {}

global a

a = int(input('Enter the number of records you want to input:\n'))

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

for i in range(a):

s['TravelID'] = int(input('Enter Travel ID:\n'))

s['From'] = input('From:\n')

s['To'] = input('To:\n')

p.dump(s, f)

def apend():

'''Adds travel data to the travel binary file'''

s = {}

global a

c = int(input('Enter the number of records you want to add:\n'))

a += c

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

for i in range(c):

s['TravelID'] = int(input('Enter Travel ID:\n'))

s['From'] = input('From:\n')

s['To'] = input('To:\n')

p.dump(s, f)

def updatetravelid():

'''Changes the values of the travel data using travel id'''

s = {}

c = int(input('Enter the Travel ID:\n'))

d = input('Enter the changed From:\n')

e = input('Enter the changed To:\n')

with open('travel.dat', 'rb+') as f:

for i in range(a):

r = f.tell() # This should always be before load

s = p.load(f)

if s['TravelID'] == c:

s['From'] = d

s['To'] = e

f.seek(r)

p.dump(s, f)

def display():

'''Displays the data on the binary file travel'''

s = {}

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

for i in range(a):

s = p.load(f)

print(s)

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.\n'''))

if n == 1:

bina()

elif n == 2:

apend()

elif n == 3:

updatetravelid()

elif n == 4:

display()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.

1

Enter the number of records you want to input:

1

Enter Travel ID:

1001

From:

Bangalore

To:

Chennai

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.

2

Enter the number of records you want to add:

1

Enter Travel ID:

1002

From:

Chennai

To:

Kolkata

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.

4

{'TravelID': 1001, 'From': 'Bangalore', 'To': 'Chennai'}

{'TravelID': 1002, 'From': 'Chennai', 'To': 'Kolkata'}

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.

3

Enter the Travel ID:

1002

Enter the changed From:

Chennai

Enter the changed To:

Bangalore

Do you want to try again? (Y/N)

y

Enter 1 to create a binary file named Travel and add Travel id, From and To.

Enter 2 to add data to the binary file.

Enter 3 to update a record based on the Travel ID.

Enter 4 to read contents of the file.

4

{'TravelID': 1001, 'From': 'Bangalore', 'To': 'Chennai'}

{'TravelID': 1002, 'From': 'Chennai', 'To': 'Bangalore'}

Do you want to try again? (Y/N)

n

**Program 12:**

Write a menu driven program in Python:

1. Define a function to write the following data into a **CSV file**:

* Roll no
* Name of student
* Mark in Sub1
* Mark in sub2
* Mark in sub3
* Mark in sub4
* Mark in sub5

**Perform following operations on the CSV file:**

1. Define a function to read the CSV file and calculate total and percentage for each student.
2. Define a function to display the name of student if in any subject marks are greater than 80% (Assume marks are out of 100)

**Code:**

import csv

def writeintocsv():

'''Writes the data into the csv file'''

with open('student.csv', 'w', newline='') as f:

w = csv.writer(f)

a = int(input('Enter the no. of records you want to enter:\n'))

for i in range(a):

r = int(input('Enter Roll No.:\n'))

am = input('Enter Name:\n')

m1 = int(input('Enter marks of Subject 1:\n'))

m2 = int(input('Enter marks of Subject 2:\n'))

m3 = int(input('Enter marks of Subject 3:\n'))

m4 = int(input('Enter marks of Subject 4:\n'))

m5 = int(input('Enter marks of Subject 5:\n'))

l = [r, am, m1, m2, m3, m4, m5]

w.writerow(l)

def totalpercen():

'''Calculates Total and Percentage of each student'''

total = 0

with open('student.csv') as f:

c = csv.reader(f)

for i in c:

for j in i[2:]:

total += int(j)

print(

f'The Total and Percentage of {i[1]} is {total} and {total/5} respectively.')

total = 0

def display():

'''Displays the name of the student who have any marks that are grater than 80'''

test = True

with open('student.csv') as f:

c = csv.reader(f)

for i in c:

for j in i[2:]:

a = int(j)

if a > 80:

print(

f'{i[1]} has scored more than 80 in Subject {i.index(j)-1}.')

test = False

if test:

print('None')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to write onto a csv file the Roll No., Name, Marks for Subjects 1,2,3,4 & 5.

Enter 2 to calculate Total and Percentage of each Student.

Enter 3 to display the name of the Student who have any marks that are grater than 80.\n'''))

if n == 1:

writeintocsv()

elif n == 2:

totalpercen()

elif n == 3:

display()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to write onto a csv file the Roll No., Name, Marks for Subjects 1,2,3,4 & 5.

Enter 2 to calculate Total and Percentage of each Student.

Enter 3 to display the name of the Student who have any marks that are grater than 80.

1

Enter the no. of records you want to enter:

3

Enter Roll No.:

1

Enter Name:

Andy

Enter marks of Subject 1:

56

Enter marks of Subject 2:

87

Enter marks of Subject 3:

45

Enter marks of Subject 4:

73

Enter marks of Subject 5:

48

Enter Roll No.:

2

Enter Name:

Bella

Enter marks of Subject 1:

67

Enter marks of Subject 2:

75

Enter marks of Subject 3:

78

Enter marks of Subject 4:

65

Enter marks of Subject 5:

56

Enter Roll No.:

3

Enter Name:

Caroline

Enter marks of Subject 1:

56

Enter marks of Subject 2:

76

Enter marks of Subject 3:

56

Enter marks of Subject 4:

81

Enter marks of Subject 5:

76

Do you want to try again? (Y/N)

y

Enter 1 to write onto a csv file the Roll No., Name, Marks for Subjects 1,2,3,4 & 5.

Enter 2 to calculate Total and Percentage of each Student.

Enter 3 to display the name of the Student who have any marks that are grater than 80.

2

The Total and Percentage of Andy is 309 and 61.8 respectively.

The Total and Percentage of Bella is 341 and 68.2 respectively.

The Total and Percentage of Caroline is 345 and 69.0 respectively.

Do you want to try again? (Y/N)

y

Enter 1 to write onto a csv file the Roll No., Name, Marks for Subjects 1,2,3,4 & 5.

Enter 2 to calculate Total and Percentage of each Student.

Enter 3 to display the name of the Student who have any marks that are grater than 80.

3

Andy has scored more than 80 in Subject 2.

Caroline has scored more than 80 in Subject 4.

Do you want to try again? (Y/N)

n

**Program 13:**

Write a menu driven program in Python:

1. Define a function to write the following data into a CSV file “**emp.CSV**”

* empid
* empName
* Salary
* Department

1. Define a function to search the record from “**emp.CSV**” file based on empid. If record found display the record, otherwise display appropriate message.

**Code:**

import csv

def writecsvemp():

'''Creates and writes onto a csv file named emp which stores Employee ID,

Employee name, Employee's Salary and Employee's Department'''

a = int(input('Enter the no. of records you want to input:\n'))

with open('emp.csv', 'w', newline='') as f:

w = csv.writer(f)

for i in range(a):

empid = int(input('Enter Employee ID:\n'))

empam = input('Enter Name of Employee:\n')

sal = input('Enter Salary of Employee:\n')

dept = input('Enter the Name of the Department:\n')

l = [empid, empam, sal, dept]

w.writerow(l)

def searchemp():

'''Searches for a record in emp.csv and returns it if found or

gives an error message if not'''

a = int(input('Enter Employee ID:\n'))

e = True

with open('emp.csv') as f:

c = csv.reader(f)

for i in c:

if a == int(i[0]):

print(i)

e = False

if e:

print('The Record was not found.')

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to create and write onto a csv file named emp.csv which stores Employee ID,

Employee name, Employee's Salary and Employee's Department.

Enter 2 to search a record using the Employee ID.\n'''))

if n == 1:

writecsvemp()

elif n == 2:

searchemp()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to create and write onto a csv file named emp.csv which stores Employee ID,

Employee name, Employee's Salary and Employee's Department.

Enter 2 to search a record using the Employee ID.

1

Enter the no. of records you want to input:

3

Enter Employee ID:

1

Enter Name of Employee:

Ayan

Enter Salary of Employee:

20000

Enter the Name of the Department:

IT

Enter Employee ID:

3

Enter Name of Employee:

Betty

Enter Salary of Employee:

25000

Enter the Name of the Department:

HR

Enter Employee ID:

4

Enter Name of Employee:

Calvin

Enter Salary of Employee:

30000

Enter the Name of the Department:

SysOP

Do you want to try again? (Y/N)

y

Enter 1 to create and write onto a csv file named emp.csv which stores Employee ID,

Employee name, Employee's Salary and Employee's Department.

Enter 2 to search a record using the Employee ID.

2

Enter Employee ID:

2

The Record was not found.

Do you want to try again? (Y/N)

y

Enter 1 to create and write onto a csv file named emp.csv which stores Employee ID,

Employee name, Employee's Salary and Employee's Department.

Enter 2 to search a record using the Employee ID.

4

Wrong Input!

Try Again!

Do you want to try again? (Y/N)

y

Enter 1 to create and write onto a csv file named emp.csv which stores Employee ID,

Employee name, Employee's Salary and Employee's Department.

Enter 2 to search a record using the Employee ID.

2

Enter Employee ID:

4

['4', 'Calvin', '30000', 'SysOP']

Do you want to try again? (Y/N)

n

**Program 14:**

A list contains the following record of a Hostel:

**[Hostel\_No, Total\_Students, Total Rooms]**

Write a menu driven program, with the following user defined functions to perform given operations on the stack named **‘Hostel’**:

1. **Push\_element( )** - To push an object containing Hostel\_No and Total  Students along with Total Rooms to the stack
2. **Pop\_element( )** - To pop the objects from the stack and display them. Also, display “Stack Empty” when there are no elements in the stack.

**For example:**

  If the lists of Hostel details are:

**[1, 2000, 1000]**

**[2, 1500, 800]**

**[3, 5000, 2000]**

 The output should be:

**[3, 5000, 2000]**

**[2, 1500, 800]**

**[1, 2000, 1000]**

**Stack Empty**

1. Implement all the stack operations.

**Code:**

hostel = []

def isEmpty(s):

'''Checks if the stack is empty or not'''

if len(s) == 0:

return True

return False

def push\_element(l):

'''Adds the element to stack'''

hostel.append(l)

def pop\_element(s):

'''Checks if the stack is empty or not and returns a popped value from the stack'''

if isEmpty(s):

return 'Stack is Empty.'

return s.pop()

def peek(s):

'''Returns the topmost element from the stack'''

if isEmpty(s):

return 'Stack is Empty.'

return s[-1]

def display(s):

'''Displays the entire stack'''

if isEmpty(s):

print('Stack is Empty.')

return

for i in s[::-1]:

print(i)

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.\n'''))

if n == 1:

a = 'Yy'

while a in 'Yy':

ht = int(input('Enter Hostel Number:\n'))

ts = int(input('Enter Total Students:\n'))

tr = int(input('Enter Total Rooms:\n'))

a = input('Do you want to add more records? (Y/N)\n')

l = [ht, ts, tr]

push\_element(l)

elif n == 2:

print(pop\_element(hostel))

elif n == 3:

print(peek(hostel))

elif n == 4:

display(hostel)

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.

1

Enter Hostel Number:

1

Enter Total Students:

50

Enter Total Rooms:

27

Do you want to add more records? (Y/N)

y

Enter Hostel Number:

2

Enter Total Students:

30

Enter Total Rooms:

17

Do you want to add more records? (Y/N)

n

Do you want to try again? (Y/N)

y

Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.

3

[2, 30, 17]

Do you want to try again? (Y/N)

y

Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.

4

[2, 30, 17]

[1, 50, 27]

Do you want to try again? (Y/N)

y

Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.

2

[2, 30, 17]

Do you want to try again? (Y/N)

y

Enter 1 to push records of Hostel No., Total Students along with Total Rooms to a stack.

Enter 2 to Remove an element from the stack.

Enter 3 to Peek the stack.

Enter 4 to Display the stack.

4

[1, 50, 27]

Do you want to try again? (Y/N)

n

**Program 15:**

Mr. John wants to store book details like book ID and Book Name in a dictionary format. Write a menu driven program, with separate user defined functions to perform the following operations:

1. Push the values (Name of the book) of the dictionary into a stack where book names begin with A or C.
2. Pop and display the content of the stack.

For example, if the content of Dictionary is as follows:

**Book={"B001":"C++", "B002":"Python", "B003":"Ada", "B004":"C",**

**"B005":"Java", "B006":"Oracle", "B007":"HTMl"}**

The output of the program should be : **C   Ada   C++**

1. Implement all the stack operations.

**Code:**

books = []

bookdic = {}

def isEmpty(s):

'''Checks if the stack is Empty or not.'''

if len(s) == 0:

return True

return False

def enterdictionary():

'''Stores the Book ID and Names of the Books in a dictionary.'''

e = eval(

input('Enter a dictionary containing Book ID And Book Name:\n'))

bookdic.update(e)

def pushvalues():

'''Adds the names of the Books starting with A or C to a stack.'''

for i in list(bookdic.values()):

if i[0] in 'CcAa':

books.append(i)

bookdic.clear()

def popvalues():

'''Removes and Returns a value from the stack.'''

if isEmpty(books):

return 'Stack Empty'

return books.pop()

def peekvalues():

'''Returns the topmost value from the stack.'''

if isEmpty(books):

return 'Stack Empty'

return books[-1]

def display():

'''Displays the whole stack from the top.'''

if isEmpty(books):

print('Stack Empty')

for i in books[::-1]:

print(i)

d = 'Yy'

while d in 'Yy':

n = int(input('''Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.\n'''))

if n == 1:

a = 'Yy'

while a in 'Yy':

enterdictionary()

a = input('Do you want to add more records? (Y/N)\n')

elif n == 2:

pushvalues()

elif n == 3:

print(popvalues())

elif n == 4:

print(peekvalues())

elif n == 5:

display()

else:

print('Wrong Input!\nTry Again!')

d = input('Do you want to try again? (Y/N)\n')

**Output:**

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

1

Enter a dictionary containing Book ID And Book Name:

{'B001':'C++'}

Do you want to add more records? (Y/N)

y

Enter a dictionary containing Book ID And Book Name:

{'B002':'Avalanches'}

Do you want to add more records? (Y/N)

y

Enter a dictionary containing Book ID And Book Name:

{'B003':'Huckleberry Fin'}

Do you want to add more records? (Y/N)

n

Do you want to try again? (Y/N)

y

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

2

Do you want to try again? (Y/N)

y

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

5

Avalanches

C++

Do you want to try again? (Y/N)

y

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

4

Avalanches

Do you want to try again? (Y/N)

y

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

3

Avalanches

Do you want to try again? (Y/N)

y

Enter 1 to store book details like Book ID & Book Name in a dictionary format.

Enter 2 to Push Names of the Books which start with A or C to stack.

Enter 3 to Remove an element from the stack.

Enter 4 to Peek the stack.

Enter 5 to Display the stack.

5

C++

Do you want to try again? (Y/N)

n