PRACTICAL – 10

Aim: Program to pass two list of integers in a function that will create third list that will contain all odd numbers of both given list on its left side and all the even numbers both given list on its right side.

Code:

1 comb.py - C:\Users\Gurvir\Desktop\1 comb.py (3.12.1)

File Edit Format Run Options Window Help

```
def separate(list1, list2):
    combined = list1 + list2
    odd=[]
    even=[]
    for num in combined:
        if num % 2 != 0:
            odd.append(num)
        elif num % 2 == 0:
            even.append(num)
        else:
            break
    result = odd + even
    return result

list1 = eval(input('Enter values for list 1: '))
list2 = eval(input('Enter values for list 2: '))

result = separate(list1, list2)
print("Result List:", result)
```

Output:

```
= RESTART: C:\Users\Gurvir\Desktop\1 comb.py
Enter values for list 1: [23,45,12,18,13,89,65]
Enter values for list 2: [1,69,47,36,55,420,8]
Result List: [23, 45, 13, 89, 65, 1, 69, 47, 55, 12, 18, 36, 420, 8]
>>>
```

PRACTICAL – 11

Aim: Program to find frequency of a word in a sentence using functions and dictionary.

Code:

2 freq.py - C:\Users\Gurvir\Desktop\2 freq.py (3.12.1)

File Edit Format Run Options Window Help

```
def char(data):
    d = {}
    for ch in data:
        if ch not in d:
            d[ch] = data.count(ch)
    for key, value in d.items():
        print(key, ":", value)

def word(data):
    d = {}
    words = data.split()
    for word in words:
        if word not in d:
            d[word] = words.count(word)
    for key, value in d.items():
        print(key, ":", value)

c = 'y'
while c == 'y':
    print("-------------MENU-------------")
    print("1. Count characters")
    print("2. Count words")
    ch = eval(input("Enter choice: "))
    if ch == 1:
        line = input("Enter a sentence: ")
        char(line)
    elif ch == 2:
        line = input("Enter a sentence: ")
        word(line)
    else:
        print("Invalid choice")
    c = input("Want to continue? (y/n): ")
```

Output:

```
-------------MENU-------------
1. Count characters
2. Count words
Enter choice: 1
Enter a sentence: How AMD Is Advancing the 30x25 Energy Efficiency Goal in High-Performance Computing and AI
H : 2
o : 4
w : 1
  : 13
A : 3
M : 1
D : 1
I : 2
s : 1
d : 2
v : 1
a : 4
n : 8
c : 4
i : 6
g : 4
t : 2
h : 2
e : 5
3 : 1
u : 1
Want to continue? (y/n): y
-------------MENU-------------
1. Count characters
2. Count words
Enter choice: 2
Enter a sentence: How AMD Is Advancing the 30x25 Energy Efficiency Goal in High-Performance Computing and AI
How : 1
AMD : 1
Is : 1
Advancing : 1
the : 1
30x25 : 1
Energy : 1
Efficiency : 1
Goal : 1
in : 1
High-Performance : 1
Computing : 1
and : 1
AI : 1
Want to continue? (y/n): n
```

PRACTICAL – 12

Aim: Program to find the longest and shortest word with their length using functions.

Code:

3 long short.py - C:\Users\Gurvir\Desktop\3 long short.py (3.12.1)

File Edit Format Run Options Window Help

```
def long():
    line = input("Enter a sentence: ")
    words = line.split()
    longest = ""
    maxlen = 0
    for word in words:
        wordlen = len(word)
        if wordlen > maxlen:
            longest = word
            maxlen = wordlen
    print("Longest word is '", longest, "' with length:", maxlen)

def short():
    line = input("Enter a sentence: ")
    words = line.split()
    shortest = ""
    minlen = 50
    for word in words:
        wordlen = len(word)
        if wordlen < minlen:
            shortest = word
            minlen = wordlen
    print("Shortest word is '", shortest, "' with length:", minlen)

c = 'y'
while c == "y":
    print("-------------MENU-------------")
    print("1. Longest word")
    print("2. Shortest word")
    ch = eval(input("Enter choice: "))
    if ch == 1:
        long()
    elif ch == 2:
        short()
    else:
        print("Invalid choice")
    c = input("Want to continue? (y/n): ")
```

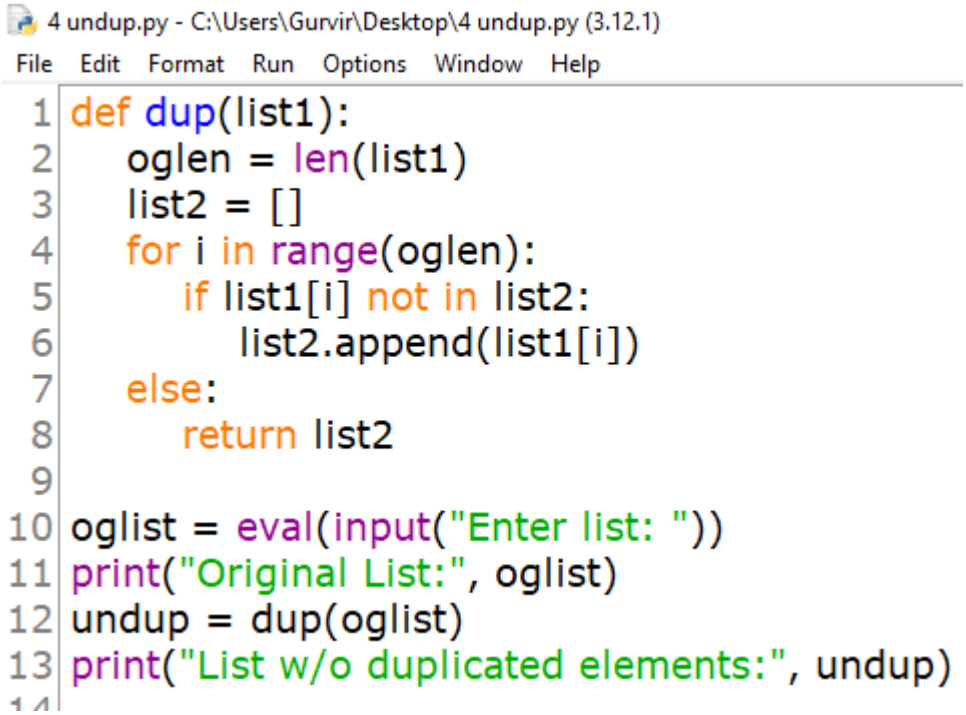
Output:

```
------------MENU-------------
1. Longest word
2. Shortest word
Enter choice: 1
Enter a sentence: AMD Pledges Extended Support: AM5 ‘Ryzen’ Desktop Socket Expec
ted to Continue Beyond 2025
Longest word is ' Extended ' with length: 8
Want to continue? (y/n): y
------------MENU-------------
1. Longest word
2. Shortest word
Enter choice: 2
Enter a sentence: AMD Radeon RX 7600 XT to Debut in January Exclusively in Custom
Variants
Shortest word is ' RX ' with length: 2
Want to continue? (y/n): n
```

PRACTICAL – 13

Aim: Program to remove duplicate elements from a list using function.

Code:



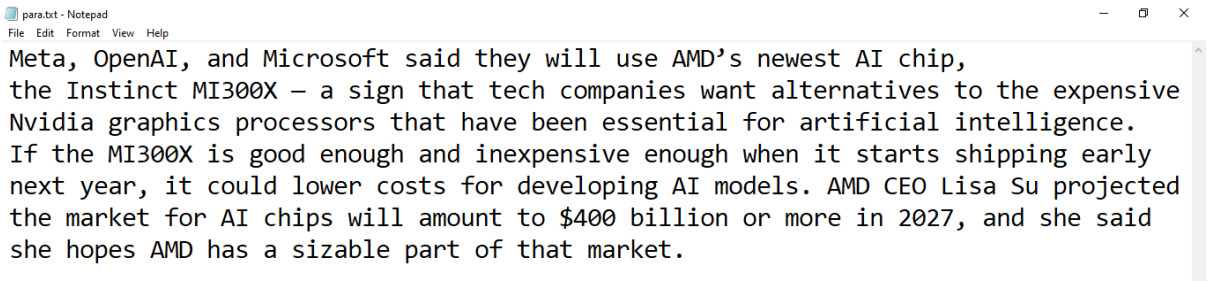
Output:

```
Enter list: [1, 2, 2, 56, 23, 12, 76, 54, 56, 76, 1]
Original List: [1, 2, 2, 56, 23, 12, 76, 54, 56, 76, 1]
List w/o duplicated elements: [1, 2, 56, 23, 12, 76, 54]
```

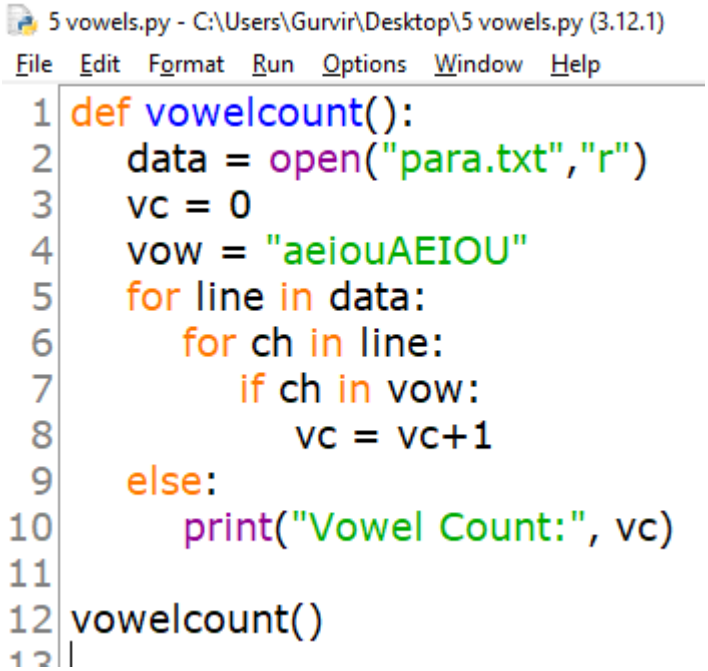
PRACTICAL – 14

Aim: Program to count number of vowels in a para.txt.

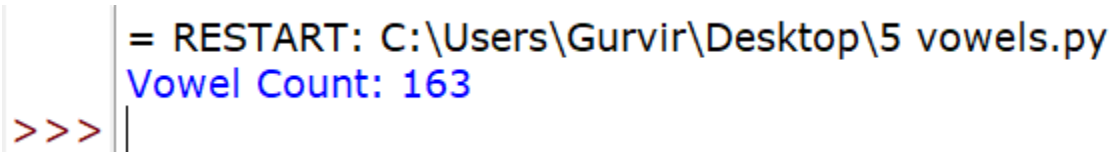
File:



Code:



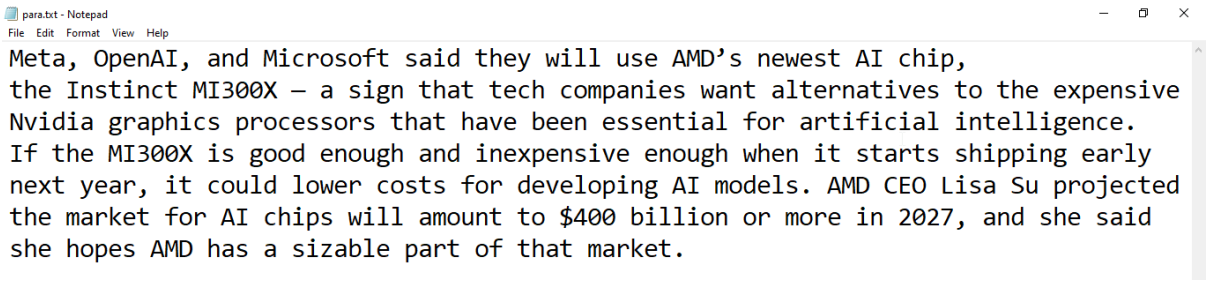
Output:



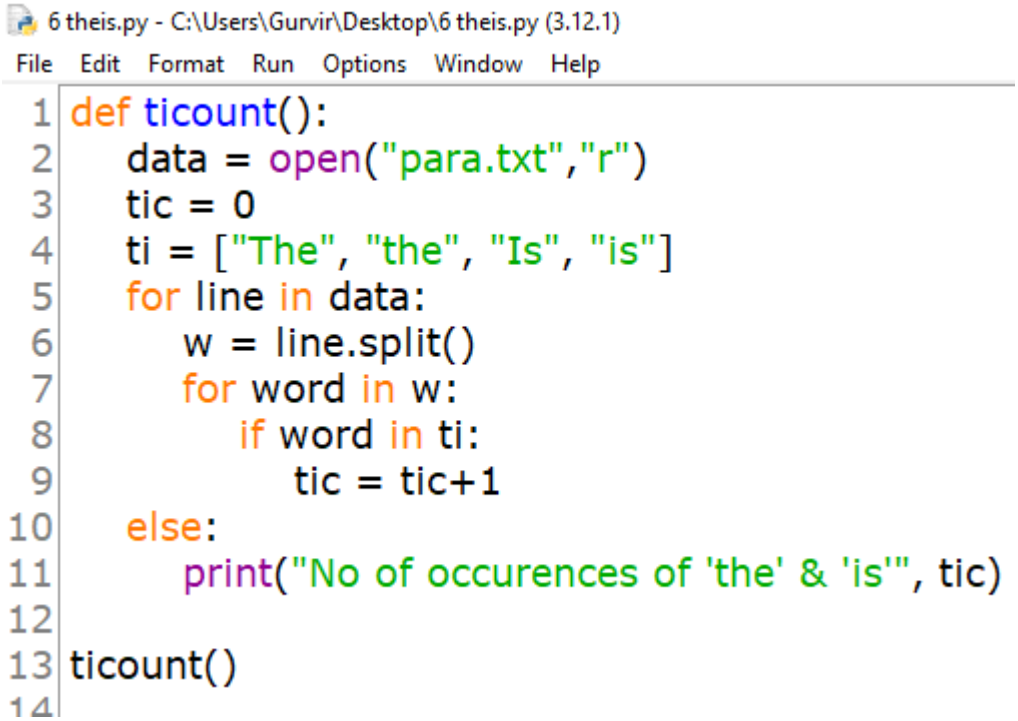
PRACTICAL – 15

Aim: Program to count number of occurrences of the word ‘the’ and ‘is’ in a para.txt.

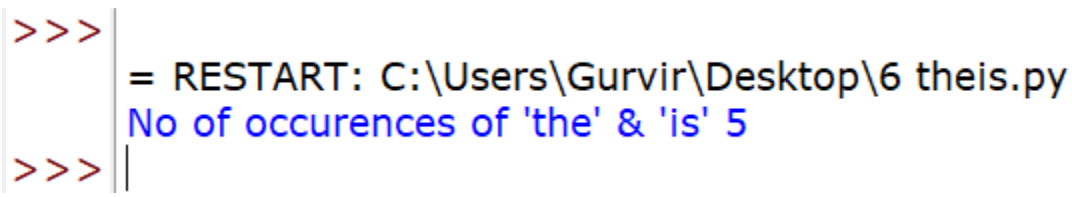
File:



Code:



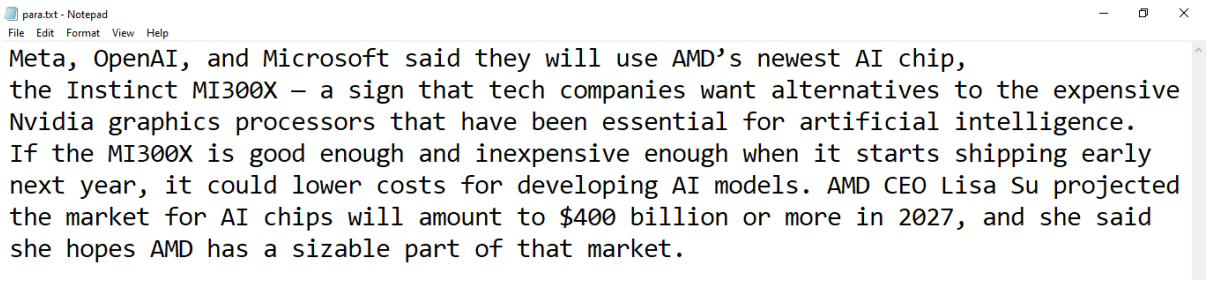
Output:



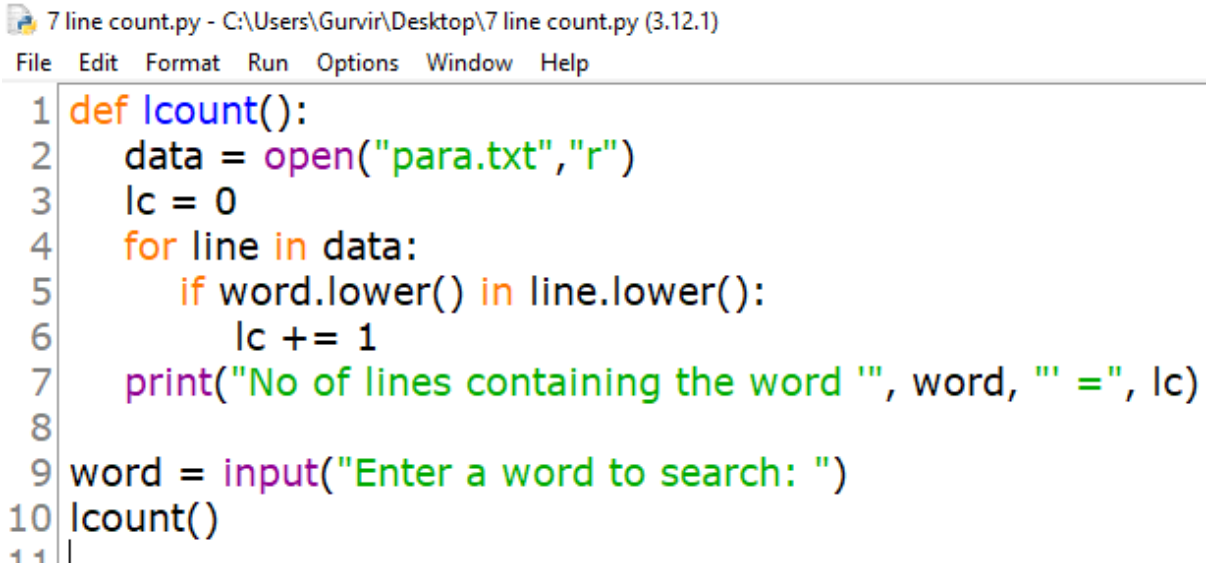
PRACTICAL – 16

Aim: Program to count the number of lines containing the word given by the user anywhere in the line in para.txt.

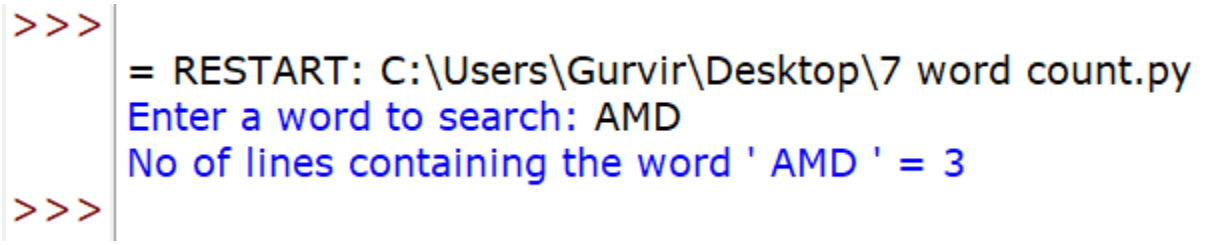
File:



Code:



Output:



PRACTICAL – 17

Aim: Program to store student data in a binary file taken from the user in the form of list and then display the record of those students whose age is more than 15 years.

Code:

```
import pickle as p

def dump():
    f = open("student1.dat", "wb")
    ch = "y"
    while ch == "y":
        rno = eval(input("Enter Roll No.: "))
        name = input("Enter Name: ")
        age = eval(input("Enter Age: "))
        strec = [rno, name, age]
        p.dump(strec, f)
        ch = input("Want to enter more records? (y/n): ")
    print("Records added to the file successfully...")
    f.close()

def load15():
    f = open("student1.dat", "rb")
    try:
        print("{0:^10}{1:<20}{2:>8}".format("R. No.", "Name", "Age"))
        print("==========================")
        while True:
            rno, name, age = p.load(f)
            if age > 15:
                print("{0:^10}{1:<20}{2:>8}".format(rno, name, age))
    except EOFError:
        pass
        .
ch = 'y'
while ch == 'y':
    print("-----------------------")
    print("         Menu          ")
    print("-----------------------")
    print("1. Create File and Add Records")
    print("2. Display records with age more than 15")
    c = eval(input("Enter your choice: "))
    if c == 1:
        dump()
    elif c == 2:
        load15()
    else:
        print("Invalid choice")
    ch = input("Continue? (y/n): ")
```

Output:

```
----------------------
        Menu
----------------------
1. Create File and Add Records
2. Display records with age more than 15
Enter your choice: 1
Enter Roll No.: 1
Enter Name: Ravi
Enter Age: 16
Want to enter more records? (y/n): y
Enter Roll No.: 2
Enter Name: Uday
Enter Age: 17
Want to enter more records? (y/n): y
Enter Roll No.: 3
Enter Name: Sohail
Enter Age: 15
Want to enter more records? (y/n): n
Records added to the file successfully...
Continue? (y/n): y

Enter your choice: 2
  R. No.  Name                  Age
==========================
    1     Ravi                  16
    2     Uday                   17
Continue? (y/n): n
```

PRACTICAL – 18

Aim: Program to store teaches data in a binary file taken from the user in the form of dictionary and then display the record of those teachers whose experience is between 5 and 10 years.

Code:

```
import pickle as p

def create():
    myfile = open("teacher1.dat", "wb+")
    ch = 'y'
    while ch == 'y':
        age = eval(input("Enter age: "))
        name = input("Enter name: ")
        exp = eval(input("Enter experience: "))
        strec = {"Age":age, "Name":name, "Experience":exp}
        p.dump(strec, myfile)
        ch = input("Want to enter more? (y/n): ")
    print("Records added successfully")
    myfile.close()

def add():
    myfile = open("teacher1.dat", "ab+")
    ch = 'y'
    while ch == 'y':
        age = eval(input("Enter age: "))
        name = input("Enter name: ")
        exp = eval(input("Enter experience: "))
        strec = {"Age":age, "Name":name, "Experience":exp}
        p.dump(strec, myfile)
        ch = input("Want to enter more? (y/n): ")
    print("Records added successfully")
    myfile.close()

```

```
def display():
    myfile = open("teacher1.dat","rb")
    print("Teachers with experience between 5 and 10 years")
    print("{0:^10} {1:<20} {2:>8}".format("Age","Name","Experience"))
    print("=============================================")
    try:
        while True:
            l = p.load(myfile)
            if l["Experience"]>5 and l["Experience"]<10:
                print("{0:^10} {1:<20} {2:>8}".format(l["Age"],l["Name"],l["Experience"]))
    except EOFError:
        pass
    myfile.close()

ch = 'y'
while ch == 'y':
    print("         MENU          ")
    print("1. Create file")
    print("2. Add record")
    print("3. Display records with condition")
    print("4. Exit")
    c = eval(input("Enter choice: "))

    if c == 1:
        create()
    elif c == 2:
        add()
    elif c == 3:
        display()
    elif c == 4:
        break
    else:
        print("Invalid choice")
```

Output:

```
         MENU
1. Create file
2. Add record
3. Display records with condition
4. Exit
Enter choice: 1
Enter age: 23
Enter name: Nikita
Enter experience: 3
Want to enter more? (y/n): y
Enter age: 45
Enter name: Ankita
Enter experience: 21
Want to enter more? (y/n): n
Records added successfully
```

```
        MENU
1. Create file
2. Add record
3. Display records with condition
4. Exit
Enter choice: 2
Enter age: 34
Enter name: Shushmita
Enter experience: 9
Want to enter more? (y/n): n
Records added successfully
        MENU
1. Create file
2. Add record
3. Display records with condition
4. Exit
Enter choice: 3
Teachers with experience between 5 and 10 years
  Age     Name              Experience
==========================================
   34     Shushmita              9
```

PRACTICAL – 19

Aim: Program to store product details in a csv file taken from user in the form of list and then display the records of those whose price is more than 100rs.

Code:

```
import csv
fields = ['Product ID', 'Product Name', 'Product Price']
file = 'product.csv'

def addrec():
    with open(file, 'w', newline='') as f:
        csvwriter = csv.writer(f)
        csvwriter.writerow(fields)
        ch = 'y'
        while ch == 'y':
            prodid = eval(input("Enter Product ID: "))
            prodname = input("Enter Product Name: ")
            prodprice = eval(input("Enter Product Price: "))
            rec = [prodid, prodname, prodprice]
            csvwriter.writerow(rec)
            ch = input("Want to enter more products? (y/n): ")
        print("CSV file created successfully")
    f.close()

def price100():
    with open(file, 'r') as f:
        data = csv.reader(f)
        next(data)
        for row in data:
            if int(row[2]) > 100:
                print(row)
            else:
                continue
```

```
    f.close()

ch = 'y'
while ch == 'y':
    print("-----------------------")
    print("         Menu          ")
    print("-----------------------")
    print("1. Create File and Add Records")
    print("2. Display records with price more than 100")
    c = eval(input("Enter choice: "))
    if c == 1:
        addrec()
    elif c == 2:
        price100()
    else:
        print('Invalid choice')
    ch = input('Continue? (y/n): ')
```

Output:

```
-----------------------
        Menu
-----------------------
1. Create File and Add Records
2. Display records with price more than 100
Enter choice: 1
Enter Product ID: 101
Enter Product Name: Toothpaste
Enter Product Price: 85
Want to enter more products? (y/n): y
Enter Product ID: 102
Enter Product Name: Ketchup
Enter Product Price: 200
Want to enter more products? (y/n): y
Enter Product ID: 103
Enter Product Name: Shampoo
Enter Product Price: 95
Want to enter more products? (y/n): y
Enter Product ID: 104
Enter Product Name: Mix Box of Sweets
Enter Product Price: 720
Want to enter more products? (y/n): n
CSV file created successfully
Continue? (y/n): y
```

```
----------------------
        Menu
----------------------
1. Create File and Add Records
2. Display records with price more than 100
Enter choice: 2
['102', 'Ketchup', '200']
['104', 'Mix Box of Sweets', '720']
Continue? (y/n): n
```

PRACTICAL – 20

Aim: Program to take student data from user in the form of list and show push and pop operation with that data in a stack.

Code:

```
stk = []

def push(stk):
    ch = 'y'
    while ch == 'y':
        print("Enter student data below:-")
        rno = eval(input("Enter Roll No.: "))
        name = input("Enter Name: ")
        marks = eval(input("Enter Marks: "))
        temp = (rno, name, marks)
        stk.append(temp)
        ch = input("Want to enter more records? (y/n): ")

def pop(stk):
    if (len(stk)) == 0:
        print("Stack is empty")
    else:
        print("Popped", stk.pop())

def display(stk):
    if (len(stk)) == 0:
        print("Stack is empty")
    else:
        for i in range(len(stk)):
            print(i+1, "-", stk[i])

```

```
ch = 'y'
while ch == 'y':
    print("         MENU          ")
    print("1. Push record")
    print("2. Pop record")
    print("3. Display stack")
    print("4. Exit")
    c = eval(input("Enter choice (1-4): "))
    if c == 1:
        push(stk)
    elif c == 2:
        pop(stk)
    elif c == 3:
        display(stk)
    elif c == 4:
        break
    else:
        print("Invalid choice")
    ch = input("Want to continue? (y/n): ")

```

Output:

```
        MENU
1. Push record
2. Pop record
3. Display stack
4. Exit
Enter choice (1-4): 1
Enter student data below:-
Enter Roll No.: 1
Enter Name: Ravi
Enter Marks: 315
Want to enter more records? (y/n): y
Enter student data below:-
Enter Roll No.: 2
Enter Name: Divyanshoo
Enter Marks: 385
Want to enter more records? (y/n): y
Enter student data below:-
Enter Roll No.: 3
Enter Name: Sohail
Enter Marks: 420
Want to enter more records? (y/n): n
Want to continue? (y/n): y

Enter choice (1-4): 2
Popped (3, 'Sohail', 420)
Want to continue? (y/n): y

Enter choice (1-4): 3
1 - (1, 'Ravi', 315)
2 - (2, 'Divyanshoo', 385)
Want to continue? (y/n): n
```