Page No.
Date

Assignment 1

class : SE IV
Batch - F6

ROLL NO: 21430
DOS : 26/9/2020

Problem statement:
In second year computer engineering class group A students play cricket, group B students play badminton and group C students play football. Write a python program using functions to compute following :- a) List of students who play both cricket and badminton b) List of students who play either cricket or badminton but not both. c) Number of students who play neither cricket nor badminton. d) Number of student who play cricket and football but not badminton.

Learning objective :-
1. To write, learn and execute simple python program
2. To learn the use of data structure, list and its operation
3. To learn use of functions in python.

Learning outcomes
1. Will be aware of executing simple python code on various editor like pycharm, eclipse etc.
2. Will be well aware of list data structure & its operations both inbuilt and user defined.
3. Write & execute well documented functions (calling, header, body closing).

S/W and H/W requirements

1. Pycharm IDE
2. Windows 64 bit

Theory -:

1. List Data structure -:

A list is a data structure in python that is mutable or changable ordered sequence of elements Each element or value that is inside of list is called an item. Lists are defined by having values betⁿ. squarebracket ([ ]).

syntax:

```
list_name = [ ]   #initialization
```

Eg. ls = [1, 2, 3, 4, 5]

2. Abstract data type (ADT)

ADT is a type for objects whose behaviour is defined by a set of value and a set of operations The defination of ADT only mentions what operations are to be performed but not how there operations will be implemented. It does not specify how data will be organized in memory and what algorithm will be implement hence it is called abstract. The process of prouding only essentials and hiding the details is known as abstraction.

syntax:

```
ADT_name:
        Data_1, Data_2
        fun_1()
        fun_2()
```

3. Control statements

1. for loop :-

It is a control flow statement for specifying iteraction, which allows code to be executed repeadtelly

syntax:

```
for i in range():
    #statements.
```

2. While loop :-

While loop is a control flow statement that allows code to be executed repeadtly based on a given boolean condition.

Syntax:

```
While (condition)
  # statements.
```

* Algorithm/pseudocode :-

1. Symmetric difference of 2 sets :-

```
ls = [ ]
s1.readset()
s2.readset()
for i in s1():
start
    for i not in s2
        l2.addelement(i)
    endif
END for
for i in s2()
Start
    for i not in s1
        l2.addelement(i)
    End if
```

```
END for
return l2
END.
```

ii) Union of 2 sets

```
Uset = []
for i in s1
    if i not in uset
        Uset.addelement(i)
    END if
END for
for i in s2
    if i not in uset
        uset.addelement(i)
    END if
END for
return Uset
END
```

iii) Intersection of s1 & s2.

```
intersection (set s1, set s2)
is = [ ]
for i in s1:
    if i is in set s2
        is.addelement(i)
    end if
END for
return intersecta
END
```

Page No.
Date

Add element (ele)
if element not in set :
addele to set :
end if
end addelement.

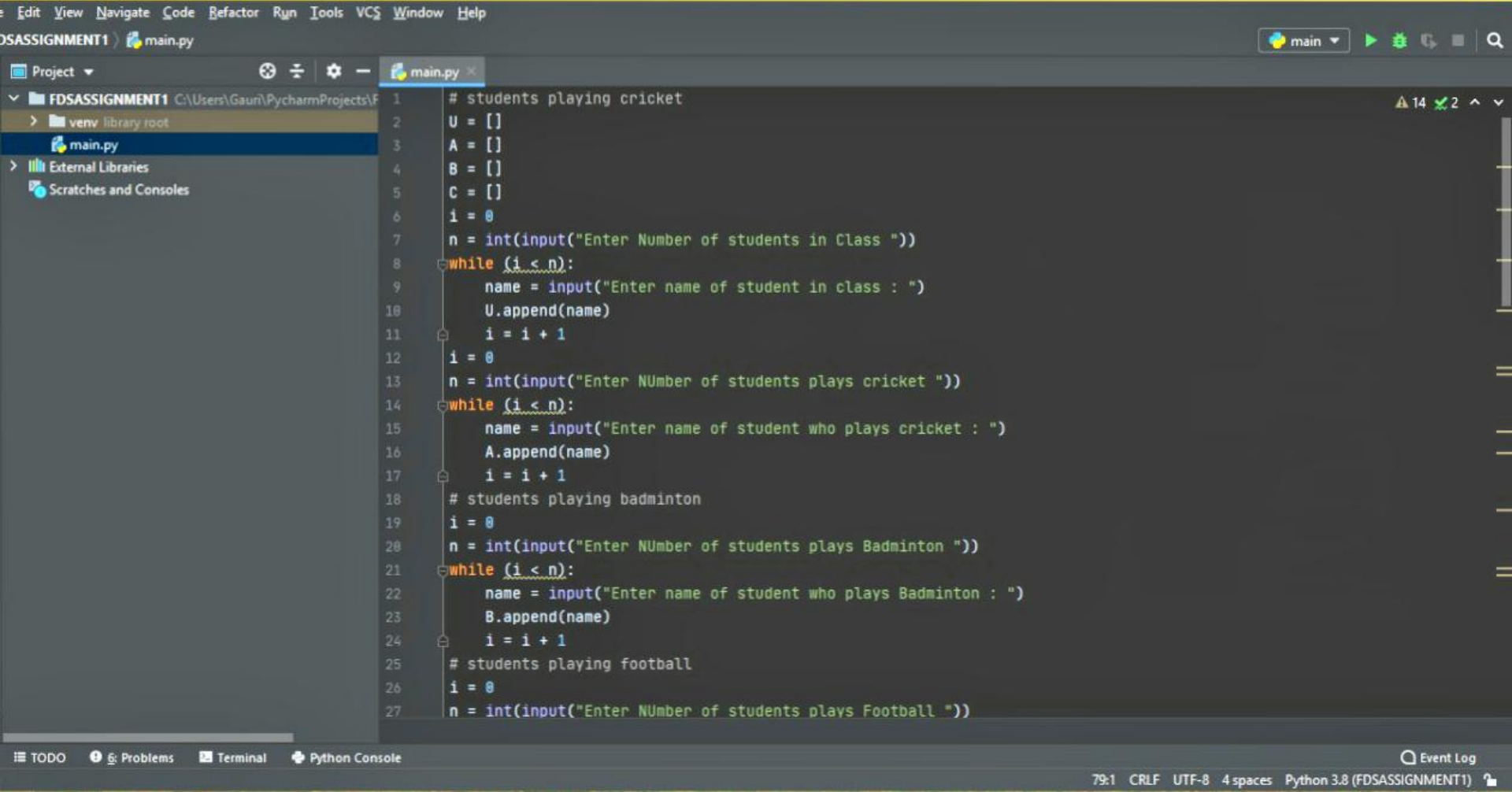
Test case :-

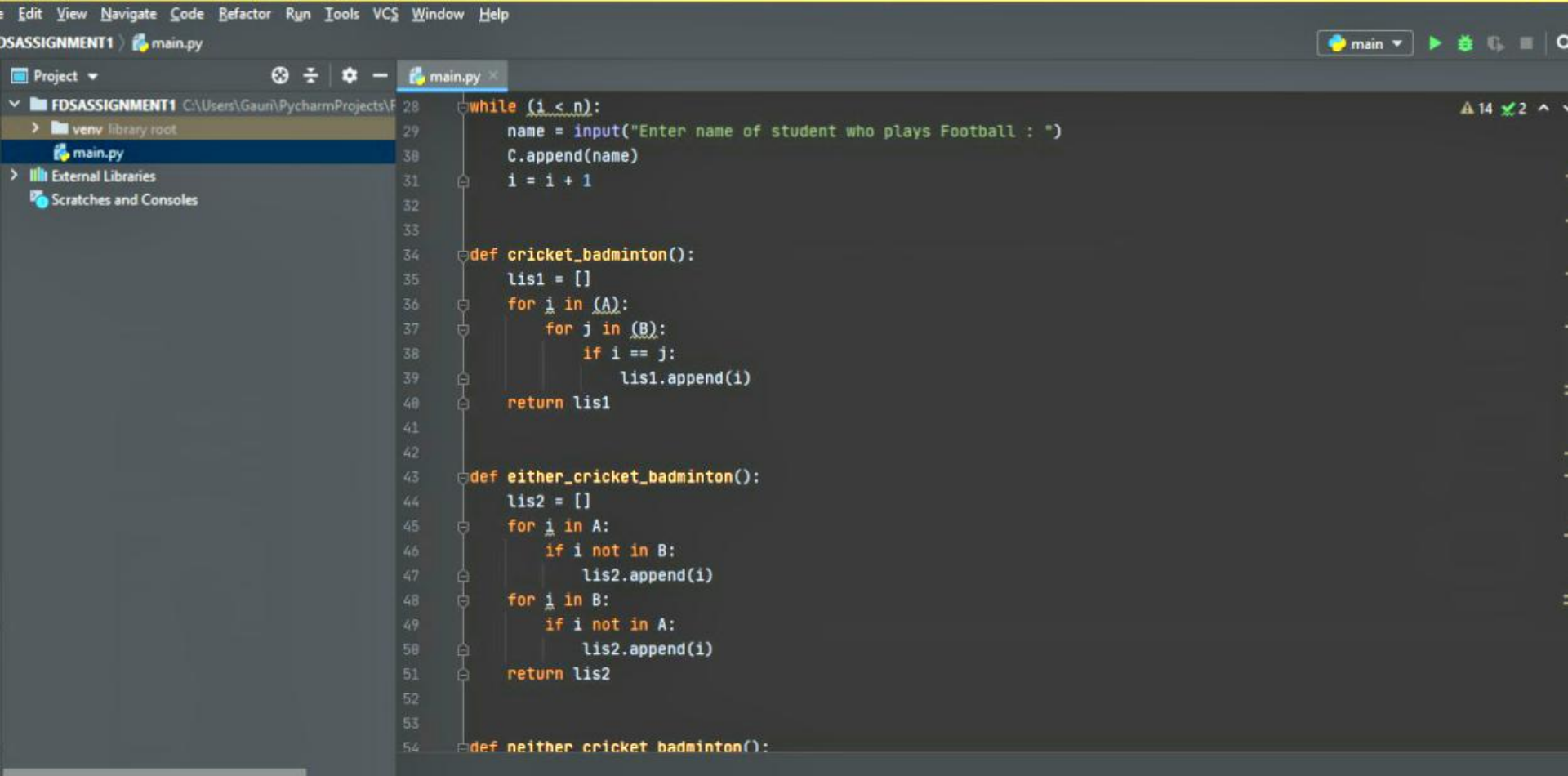
NO.	Description	Input	Expected O/P	Actual O/P	Result
1.	List of student playing cricket & badminton	S=[1,2,3,4,5] C=[1,2,3] B=[2,3,4] F=[3,4,5]	ls = [2,3]	ls=[2,3]	Pass
2	List of student not playing cricket & badminton	S=[1,2,3,4,5] C=[1,2,3] B=[2,3,4] F=[3,4,5]	ls = [5]	ls = [5]	Pass

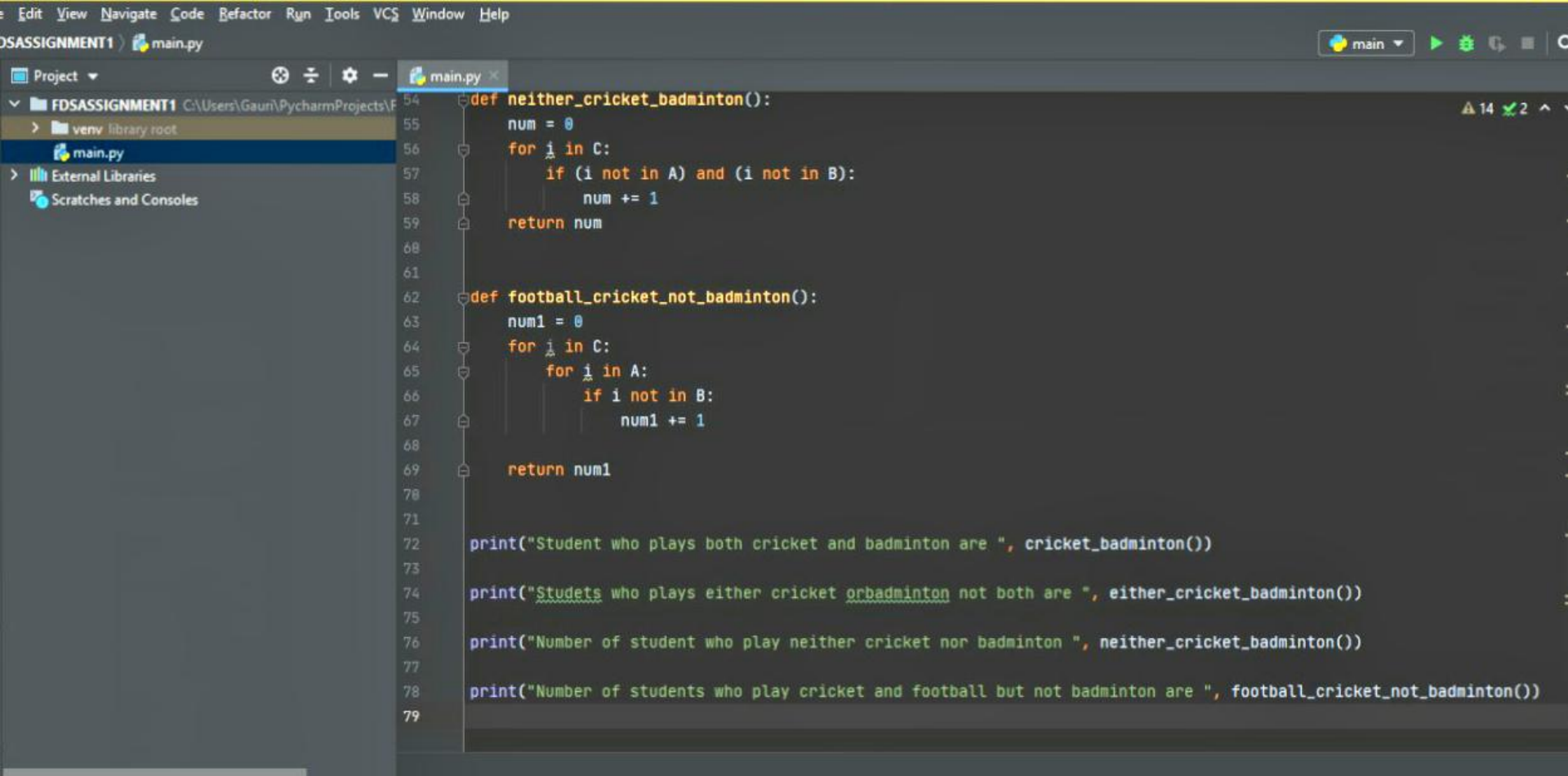
Time complexity :-

	Set	Avg. case	Worst case
1.	Union of sets	O(len(S1)+len(S2))	-
2.	Intersection of sets	O(min(S1)+len(S2))	O(len(S1)*len(S2))
3.	Symm. Diff.	O(len(S1)).	O(len(S1)*len(S2))
4.	Diff S1-S2	O(len(S1))	-
5.	Symm. Diff update	O(len(S2))	O(len(S1)*len(S2))

Conclusion :-
We learnt about writing and execting python function
Learnt to implement list data structure & its operations


Edit View Navigate Code Refactor Run Tools VCS Window Help
FDSASSIGNMENT1 main.py
main
Project
main.py
FDSASSIGNMENT1 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
14 2
# students playing cricket
U = []
A = []
B = []
C = []
i = 0
n = int(input("Enter Number of students in Class "))
while (i < n):
name = input("Enter name of student in class : ")
U.append(name)
i = i + 1
i = 0
n = int(input("Enter NUmber of students plays cricket "))
while (i < n):
name = input("Enter name of student who plays cricket : ")
A.append(name)
i = i + 1
# students playing badminton
i = 0
n = int(input("Enter NUmber of students plays Badminton "))
while (i < n):
name = input("Enter name of student who plays Badminton : ")
B.append(name)
i = i + 1
# students playing football
i = 0
n = int(input("Enter NUmber of students plays Football "))
TODO
6: Problems
Terminal
Python Console
Event Log
79:1 CRLF UTF-8 4 spaces Python 3.8 (FDSASSIGNMENT1)


Edit View Navigate Code Refactor Run Tools VCS Window Help
DSASSIGNMENT1 main.py
main
Project
main.py
FDSASSIGNMENT1 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
14 2
while (i < n):
name = input("Enter name of student who plays Football : ")
C.append(name)
i = i + 1
def cricket_badminton():
lis1 = []
for i in (A):
for j in (B):
if i == j:
lis1.append(i)
return lis1
def either_cricket_badminton():
lis2 = []
for i in A:
if i not in B:
lis2.append(i)
for i in B:
if i not in A:
lis2.append(i)
return lis2
def neither cricket badminton():


Edit View Navigate Code Refactor Run Tools VCS Window Help
DSASSIGNMENT1 main.py
main
Project
main.py
FDSASSIGNMENT1 C:\Users\Gauri\PycharmProjects\F
venv library root
main.py
External Libraries
Scratches and Consoles
14 2
def neither_cricket_badminton():
num = 0
for i in C:
if (i not in A) and (i not in B):
num += 1
return num
def football_cricket_not_badminton():
num1 = 0
for i in C:
for i in A:
if i not in B:
num1 += 1
return num1
print("Student who plays both cricket and badminton are ", cricket_badminton())
print("Studets who plays either cricket orbadminton not both are ", either_cricket_badminton())
print("Number of student who play neither cricket nor badminton ", neither_cricket_badminton())
print("Number of students who play cricket and football but not badminton are ", football_cricket_not_badminton())

```
C:\Users\Gauri\PycharmProjects\FDSASSIGNMENT1\venv\Scripts\python.exe C:/Users/Gauri/PycharmProjects/FDSASSIGNMENT1/main.py
Enter Number of students in Class 7
Enter name of student in class : ab
Enter name of student in class : ds
Enter name of student in class : kj
Enter name of student in class : ed
Enter name of student in class : aw
Enter name of student in class : gf
Enter name of student in class : yg
Enter NUmber of students plays cricket 4
Enter name of student who plays cricket : ab
Enter name of student who plays cricket : ds
Enter name of student who plays cricket : kj
Enter name of student who plays cricket : ed
Enter NUmber of students plays Badminton 2
Enter name of student who plays Badminton : ab
Enter name of student who plays Badminton : ed
Enter NUmber of students plays Football 3
Enter name of student who plays Football : aw
Enter name of student who plays Football : ab
Enter name of student who plays Football : ds
Student who plays both cricket and badminton are  ['ab', 'ed']
Studets who plays either cricket orbadminton not both are  ['ds', 'kj']
Number of student who play neither cricket nor badminton  1
Number of students who play cricket and football but not badminton are  6

Process finished with exit code 0
```