TASK4 Use various data types, List, Tuples and Dictionary in python programming

Problem 1:

Consider a list (list=[]). You can perform the following commands:

1. insert: Insert integer 47 at position 2.
2. print: Print the list.
3. remove: Delete the first occurrence of integer 4
4. append: Insert integer 21 at the end of the list.
5. sort: Sort the list.
6. pop: Pop the last element from the list.
7. reverse: Reverse the list.

Initialize your list and read in the value of followed by lines of commands where each command will be of the types listed above. Iterate through each command in order and perform the corresponding operation on your list.

Input Format

The first line contains an integer,denoting the number of commands.
Each line of the subsequent lines contains one of the commands described above.

Constraints

- The elements added to the list must be *integers*.

Output Format

For each command of type print, print the list on a new line.

Sample Input 0

```
12
insert 0 5
insert 1 10
insert 0 6
print
remove 6
append 9
append 1
sort
print
pop
reverse
print
```

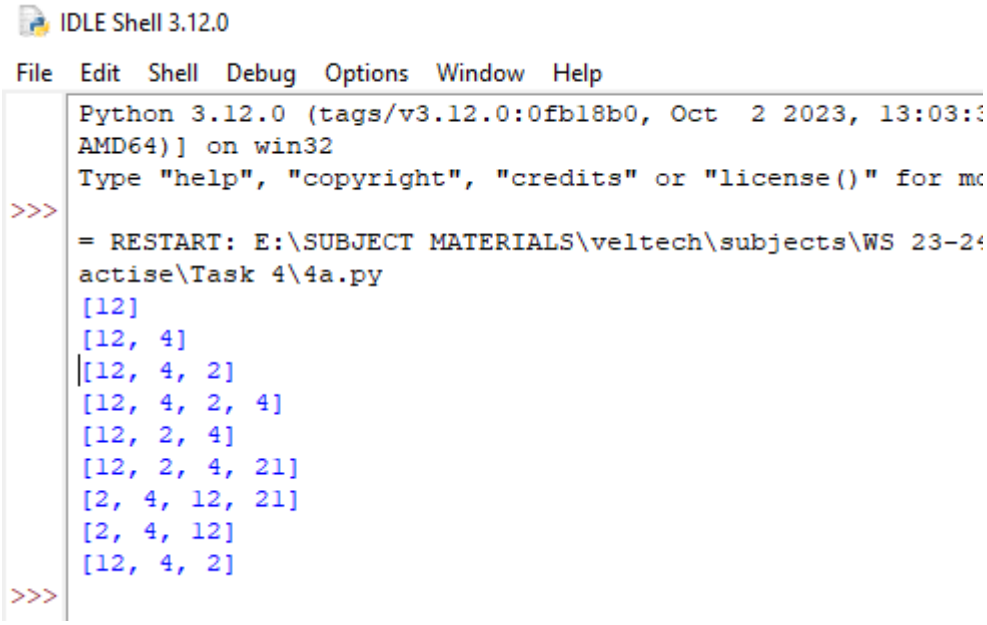
Sample Output 0

```
[6, 5, 10]
[1, 5, 9, 10]
[9, 5, 1]
```

PROGRAM

```
sample=[]

sample.insert(0,12)

print(sample)

sample.insert(1,4)

print(sample)

sample.insert(2,2)
```

```
print(sample)

sample.insert(3,4)

print(sample)

sample.remove(4)

print(sample)

sample.append(21)

print(sample)

sample.sort()

print(sample)

sample.pop()

print(sample)

sample.reverse()

print(sample)

a = list(range(10))

print(a)
```



Problem 2:

Given the names and grades for each student in a class of ***N*** students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

Note: If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

Example

records = [[“chi”, 70.0]], [“beta”, 50.0], [“alpha”, 50.0]

The ordered list of scores is **[70.0, 50.0]**, so the second lowest score is **50.0**. There are two students with that score: **[“beta”, “alpha”]**. Ordered alphabetically, the names are printed as:

alpha beta

Sample Input 0

```
5
Harry
37.21
Berry
37.21
Tina
37.2
Akriti
41
Harsh
39
```

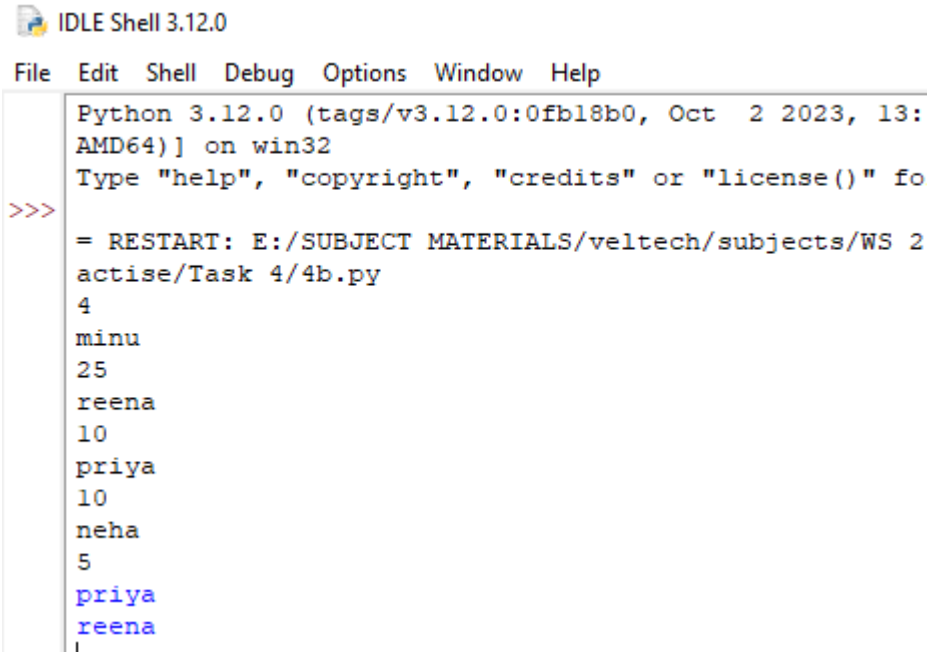
Sample Output 0

```
Berry
Harry
```

PROGRAM:

```
alist = []

for i in range(int(input())):

    name = input()

    score = float(input())

    alist.append([name, score])

second_highest = sorted(set([score for name, score in alist]))[1]  // List comprehension--→iterates

                                over each element of  alist, unpacks each element into name and
score, selects only the  score part, and constructs a new sequence (e.g., a list) containing only
the scores.

print('\n'.join(sorted([name for name, score in alist if score == second_highest])))  //List  comprehension

This joins the sorted list of names into a single string, with each name on a new line.
```

OUTPUT



Problem 3:
Given an integer, n, and n space-separated integers as input, create a tuple, t, of those n integers. Then compute and print the result of hash(t).
Input Format :
The first line contains an integer, n, denoting the number of elements in the tuple.
The second line contains n space-separated integers describing the elements in tuple t.
Output Format :
Print the result of hash(t).
Sample Input :
2
1 2

PROGRAM

```
n = int(input())

a = map(int, input().split())

t = tuple(a)
```

print(hash(t)); // is used to get the hash value of an object. We can hash only these types: Hashable types: * bool * int * long * float * string * Unicode * tuple * code object

We cannot hash of these types: Non-hashable types: * bytearray * list * set * dictionary * memoryview

OUTPUT

```
= RESTART: E:/SUBJECT MATERIALS/veltech/subjects/WS 23-24/python/lab task/lab pr
actise/Task 4/4c.py
3
2 1 4
7868426630899093470
>
```

Problem 4:
Little Malvika is very peculiar about colors. On her birthday, her mom wanted to buy balloons for decorating the house. So she asked her about her color preferences. The sophisticated little person that Malvika is, she likes only two colors — amber and brass. Her mom bought n balloons, each of which was either amber or brass in color. You are provided this information in a string s consisting of characters 'a' and 'b' only, where 'a' denotes that the balloon is amber, where 'b' denotes it being brass colored.

When Malvika saw the balloons, she was furious with anger as she wanted all the balloons of the same color. In her anger, she painted some of the balloons with the opposite color (i.e., she painted some amber ones brass and vice versa) to make all balloons appear to be the same color. As she was very angry, it took her a lot of time to do this, but you can probably show her the right way of doing so, thereby teaching her a lesson to remain calm in difficult situations, by finding out the minimum number of balloons needed to be painted in order to make all of them the same color.

Input
The first line of input contains a single integer T, denoting the number of test cases.
The first and only line of each test case contains a string s.
Output
For each test case, output a single line containing an integer — the minimum number of flips required.
Constraints
$1 \leq T \leq 100$
$1 \leq n \leq 100$, where n denotes to the length of the string s.
Example
Input:
3
ab
bb
baaba

Output:
1
0
2

PROGRAM

```
def min_paint_balloons(s):
    # Count the number of amber and brass balloons
    amber_count = s.count('a')
    brass_count = s.count('b')

    # Find the minimum count between amber and brass balloons
    min_paint = min(amber_count, brass_count)

    return min_paint

# Input the number of test cases
```

```
T = int(input("enter the total testcase"))

# Iterate through each test case
for i in range(T):
    # Input the string representing balloon colors
    s = input().strip()    #strip() is used to read a line of input from the user and remove any leading and trailing whitespace from that input
    # Calculate and print the minimum number of balloons needed to be painted
    print(min_paint_balloons(s))
```

OUTPUT