22MCA0139

RAJAT SINGH

Digital Assignment 2

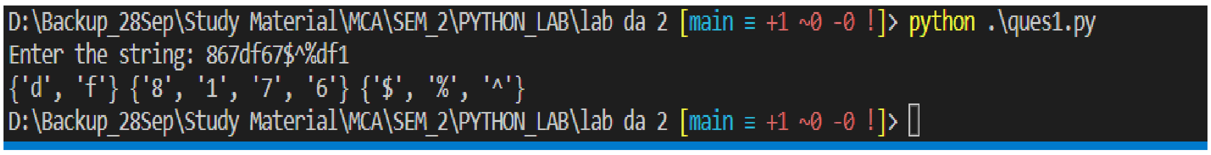
Python Programming Lab

1) **Collect a string input to variable S and create three different strings for storing characters, numbers, and special characters. Avoid repeated characters and blank spaces.**

Code:

```python
s=input("Enter the string: ")
characters=set()
numbers=set()
sp_char=set()
for i in s:
    if(i==" "):
        continue
    elif(i.isalpha()):
        characters.add(i)
    else:
        if(i.isnumeric()):
            numbers.add(i)
        else:
            sp_char.add(i)
print(characters,numbers,sp_char)
```

output:



```
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques1.py
Enter the string: 867df67$^%df1
{'d', 'f'} {'8', '1', '7', '6'} {'$', '%', '^'}
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]>
```

2) **Write your own versions of the String search methods indexOf and lastIndexOf.**

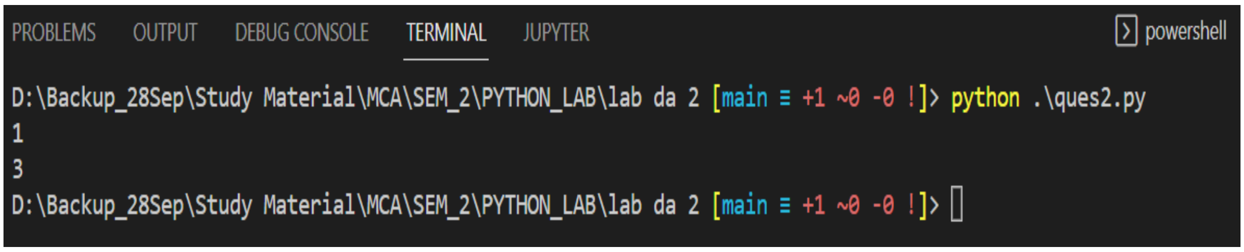
Code:

```python
def indexOf(char, string):
    for i in range(len(string)):
        if(string[i]==char):
            return i
    return -1

def lastIndexOf(char, string):
    for i in range(len(string)-1,-1,-1):
        if(string[i]==char):
            return i
    return -1

print(indexOf("a", "Rajat"))
print(lastIndexOf("a", "Rajat"))
```

Output:



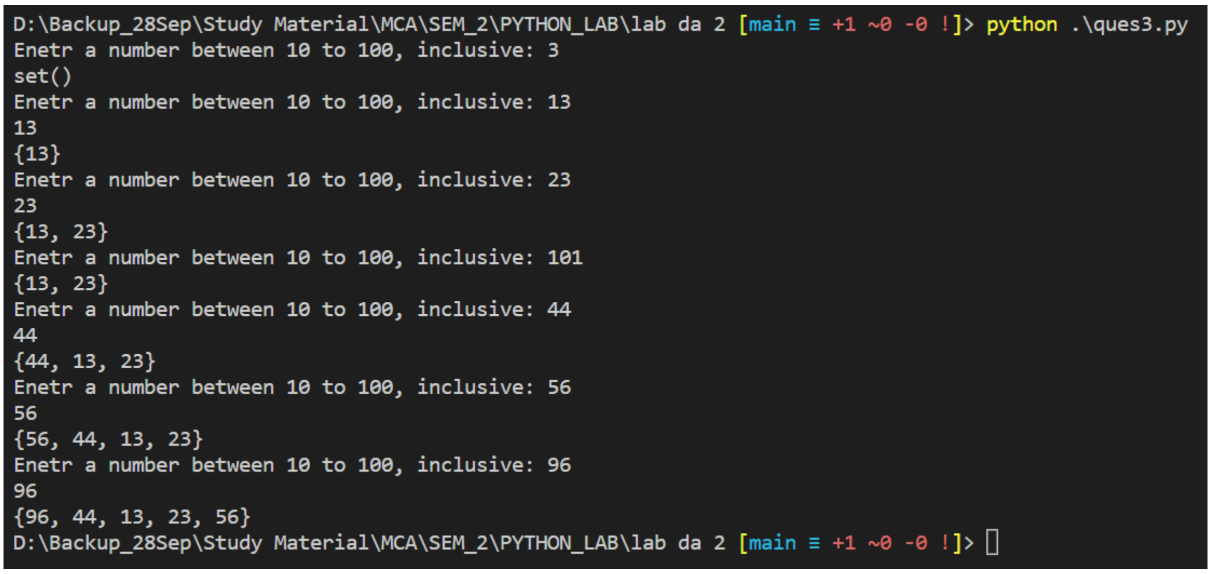
```
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques2.py
1
3
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]>
```

3) **Use a one-dimensional list to solve the following problem: Write an application that inputs five numbers, each between 10 and 100, inclusive. As each number is read, display it only if it's not a duplicate of a number already read. Provide for the "worst case," in which all five numbers are different. Use the smallest possible array to solve this problem. Display the complete set of unique values input after the user enters each new value.**

Code:

```python
arr=[]
unique=set()
while(len(unique)<5):
    x=eval(input("Enetr a number between 10 to 100, inclusive: "))
    if x not in arr and x in range(10, 101):
        print(x)
        unique.add(x)
        arr.append(x)
    print(unique)
```

Output:



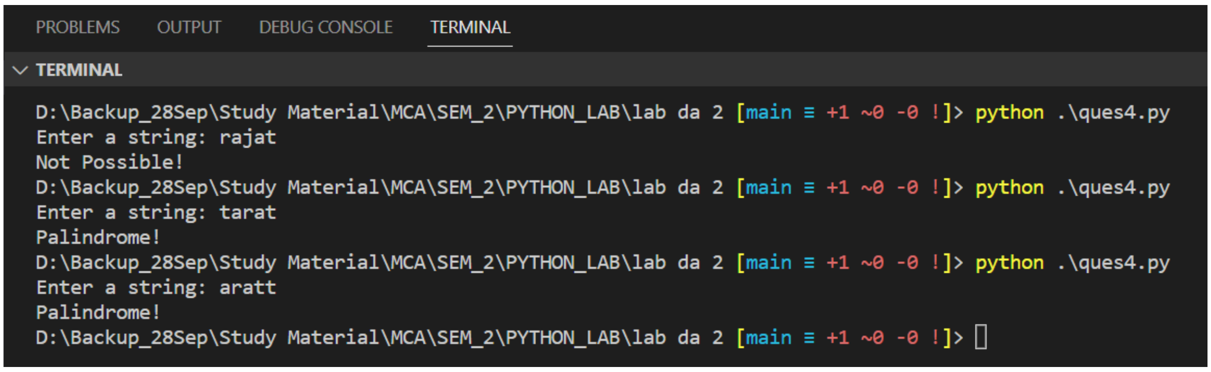
```
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques3.py
Enetr a number between 10 to 100, inclusive: 3
set()
Enetr a number between 10 to 100, inclusive: 13
13
{13}
Enetr a number between 10 to 100, inclusive: 23
23
{13, 23}
Enetr a number between 10 to 100, inclusive: 101
{13, 23}
Enetr a number between 10 to 100, inclusive: 44
44
{44, 13, 23}
Enetr a number between 10 to 100, inclusive: 56
56
{56, 44, 13, 23}
Enetr a number between 10 to 100, inclusive: 96
96
{96, 44, 13, 23, 56}
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]>
```

4) **Check whether the following string is a palindrome or not. If not, check the possibility of creating a palindrome using existing characters in the string by rearranging them. Print the palindrome string as output.**

Code:

```python
def isPalindrome(string):
    m=len(string)
    n=int(len(string)/2)
    for i in range(n):
        if(string[i]!=string[m-i-1]):
            return False
    return True
def permutations(string):
    if len(string)==1:
        return(string)
    perms=[]
    for i,c in enumerate(string):
        for perm in permutations(string[:i] + string[i+1:]):
            perms.append(c+perm)
    return perms
def allPermutation(string):
    p=permutations(string)
    for i in p:
        if(isPalindrome(i)):
            return True
    return False
s=input("Enter a string: ")
if(isPalindrome(s)):
    print("Palindrome!")
else:
    if(allPermutation(s)):
        print("Palindrome!")
    else:
        print("Not Possible!")
```

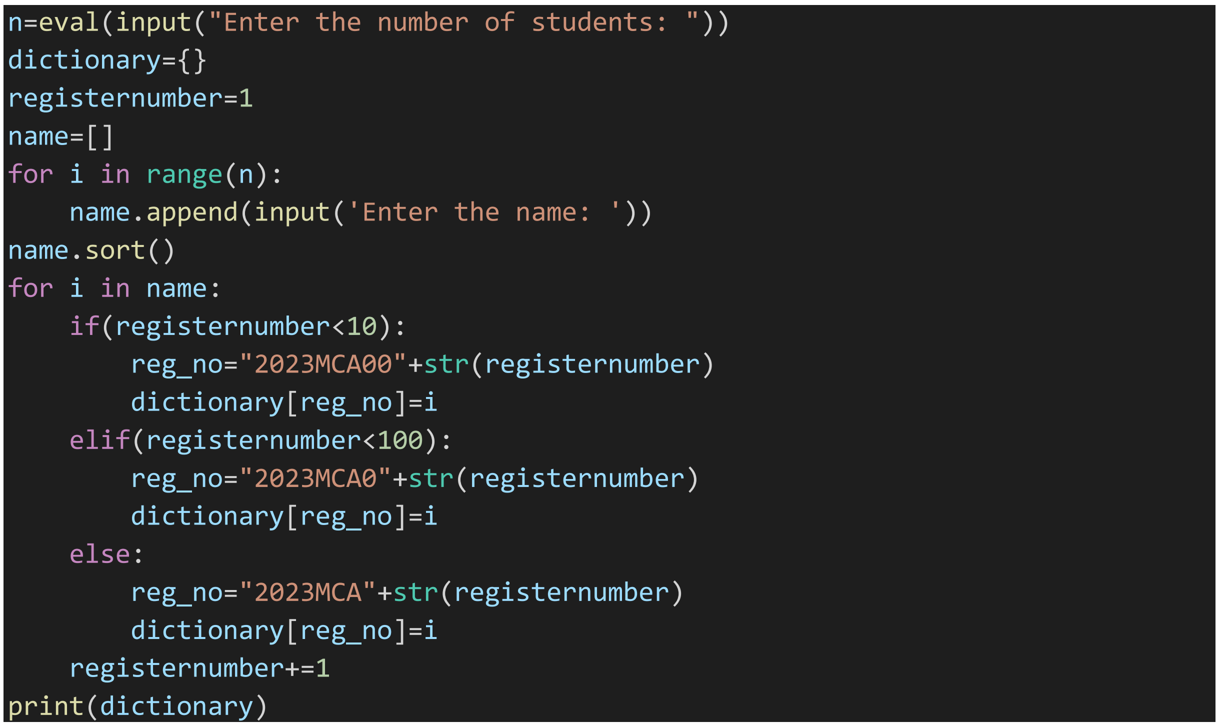
output:



```
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques4.py
Enter a string: rajat
Not Possible!
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques4.py
Enter a string: tarat
Palindrome!
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques4.py
Enter a string: aratt
Palindrome!
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]>
```

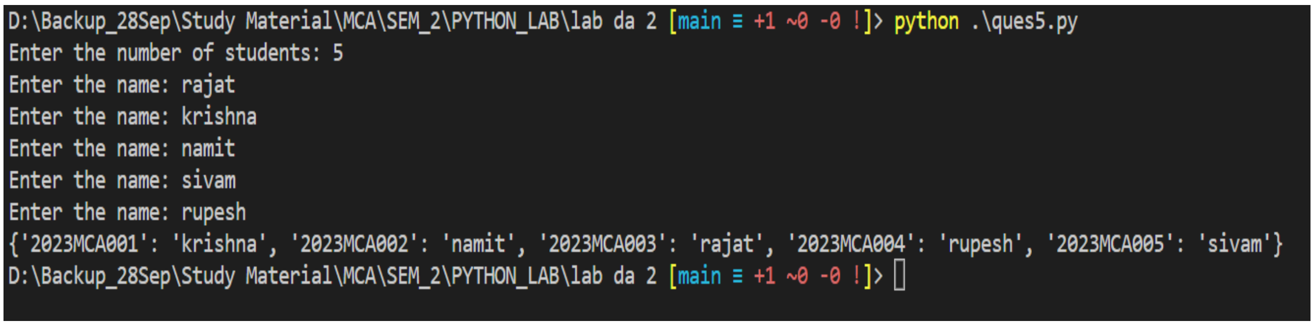
5) **Collect total number of student and their names. Assign register number for each student in alphabetical order using the template 2023MCA001. Store the name and register number of students in dictionary datatype.**

Code:



```python
n=eval(input("Enter the number of students: "))
dictionary={}
registernumber=1
name=[]
for i in range(n):
    name.append(input('Enter the name: '))
name.sort()
for i in name:
    if(registernumber<10):
        reg_no="2023MCA00"+str(registernumber)
        dictionary[reg_no]=i
    elif(registernumber<100):
        reg_no="2023MCA0"+str(registernumber)
        dictionary[reg_no]=i
    else:
        reg_no="2023MCA"+str(registernumber)
        dictionary[reg_no]=i
    registernumber+=1
print(dictionary)
```

output:



```
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]> python .\ques5.py
Enter the number of students: 5
Enter the name: rajat
Enter the name: krishna
Enter the name: namit
Enter the name: sivam
Enter the name: rupesh
{'2023MCA001': 'krishna', '2023MCA002': 'namit', '2023MCA003': 'rajat', '2023MCA004': 'rupesh', '2023MCA005': 'sivam'}
D:\Backup_28Sep\Study Material\MCA\SEM_2\PYTHON_LAB\lab da 2 [main ≡ +1 ~0 -0 !]>
```