VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

WINTER SEMESTER 2021-22

SUBJECT: PYTHON PROGRAMMING ELA(LAB)

DIGITAL ASSIGNMENT-2

Course Code: ITA6013

REGISTER NUMBER: 21MCA0017
NAME: SANJAY NAIK

Name: SANJAY NAIK
Reg No: 21MCA0017

ASSESSMENT – 2

Exercise 6

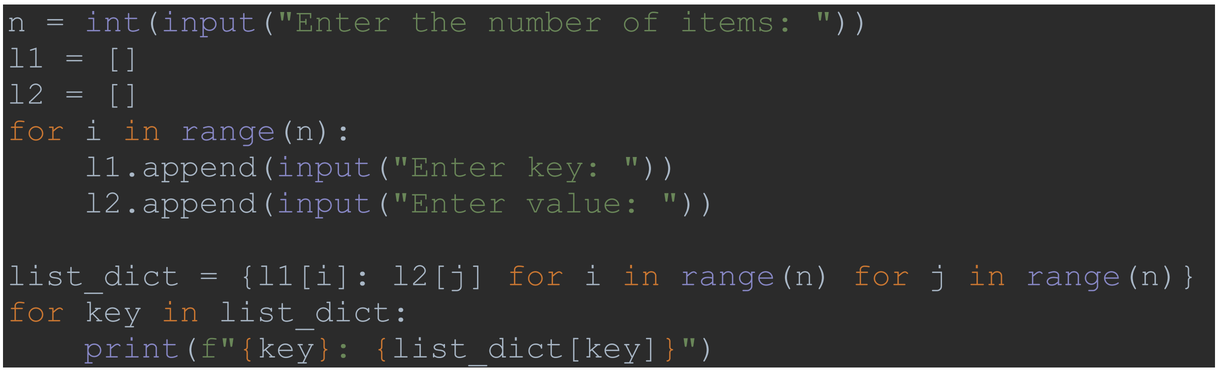
Q.1 Write a python program to store identification value and name of the MCA students in a dictionary and sort the dictionary according to its id values.

```
student_record = {}

n = int(input("Enter the number of students: "))
for i in range(n):
    id = int(input("Enter Student's Id: "))
    name = input("Enter student's name: ")
    student_record[id] = name

print("Unsorted Record")
for key in student_record:
    print(f"{key}: {student_record[key]}")

print("Sorted Record")
for i in sorted(student_record.keys()):
    print(f"{i}: {student_record[i]}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX6P1.py"
Enter the number of students: 3
Enter Student's Id: 95
Enter student's name: SANJAY
Enter Student's Id: 2
Enter student's name: SAMIKSHYA
Enter Student's Id: 40
Enter student's name: SASMITA
Unsorted Record
95: SANJAY
2: SAMIKSHYA
Sorted Record
2: SAMIKSHYA
40: SASMITA
95: SANJAY

Process finished with exit code 0
```

Q.2 Write a python program to accept two lists from the users and convert it into dictionary.



```
n = int(input("Enter the number of items: "))
l1 = []
l2 = []
for i in range(n):
    l1.append(input("Enter key: "))
    l2.append(input("Enter value: "))

list_dict = {l1[i]: l2[j] for i in range(n) for j in range(n)}
for key in list_dict:
    print(f"{key}: {list_dict[key]}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX6P2.py"
Enter the number of items: 3
Enter key: India
Enter value: New Delhi
Enter key: Spain
Enter value: Madrid
Enter key: UK
Enter value: London
India: London
Spain: London
UK: London

Process finished with exit code 0
```

Q.3 Write a python program to get the number of students from the user. For each student get name and total marks. Bases on the marks obtained from the student assign the grade to the student and store it into a dictionary and display it. Further display the grade associated with the students and counts the number of each grade.

```python
n = int(input("Enter the number of students: "))
student_marks = {}
student_grades = {}
grade_list = []

for i in range(n):
    name = input("Enter Student's name: ")
    marks = int(input("Enter Student's marks: "))
    student_marks[name] = marks

print("Student Marks: ")
for name in student_marks:
    print(f"{name}: {student_marks[name]}")

for name in student_marks:
    if student_marks[name] > 90:
        student_grades[name] = "E"
    elif 80 <= student_marks[name] < 90:
        student_grades[name] = "A"
    elif 70 <= student_marks[name] < 80:
        student_grades[name] = "B"
    elif 60 <= student_marks[name] < 70:
        student_grades[name] = "C"
    elif 50 <= student_marks[name] < 60:
        student_grades[name] = "D"
    else:
        student_grades[name] = "F"

print("Student Grades")
for name in student_grades:
    grade_list.append(student_grades[name])
    print(f"{name}: {student_grades[name]}")

for grade in grade_list:
    print(f"Number of {grade} Grade: {grade_list.count(grade)}.")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe "E:/Python 2022/EX6P3.py"
Enter the number of students: 3
Enter Student's name: SANJAY
Enter Student's marks: 83
Enter Student's name: KING
Enter Student's marks: 69
Enter Student's name: SAM
Enter Student's marks: 49
Student Marks:
SAM: 83
SANJAY: 69
KING: 49
Student Grades
SAM: A
SANJAY: C
KING: F
Number of A Grade: 1.
Number of C Grade: 1.
Number of F Grade: 1.

Process finished with exit code 0
```

Q.4 Write a python program to map the name of the country with its capital. Further swap the desired elements of the dictionary based on its key value.

```
country_dict = {}

n = int(input("Enter the number of countries: "))
for i in range(n):
    country = input("Enter the country name: ")
    capital = input("Enter capital: ")
    country_dict[country] = capital

for country in country_dict:
    print(f"{country}: {country_dict[country]}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX6P4.py"
Enter the number of countries: 3
Enter the country name: India
Enter capital: New Delhi
Enter the country name: Spain
Enter capital: Madrid
Enter the country name: UK
Enter capital: London
India: New Delhi
Spain: Madrid
UK: London

Process finished with exit code 0
```

Q.5 You are given some information about N people. Each person has a first name, last name, age and sex. Print their names in a specific format sorted by their age in ascending order i.e. the youngest person's name should be printed first. For two people of the same age, print them in the order of their input. For *Henry Davids*, the output should be:

Mr. Henry Davids

For *Mary George*, the output should be:

Ms. Mary George

Input Format

The first line contains the integer N, the number of people.

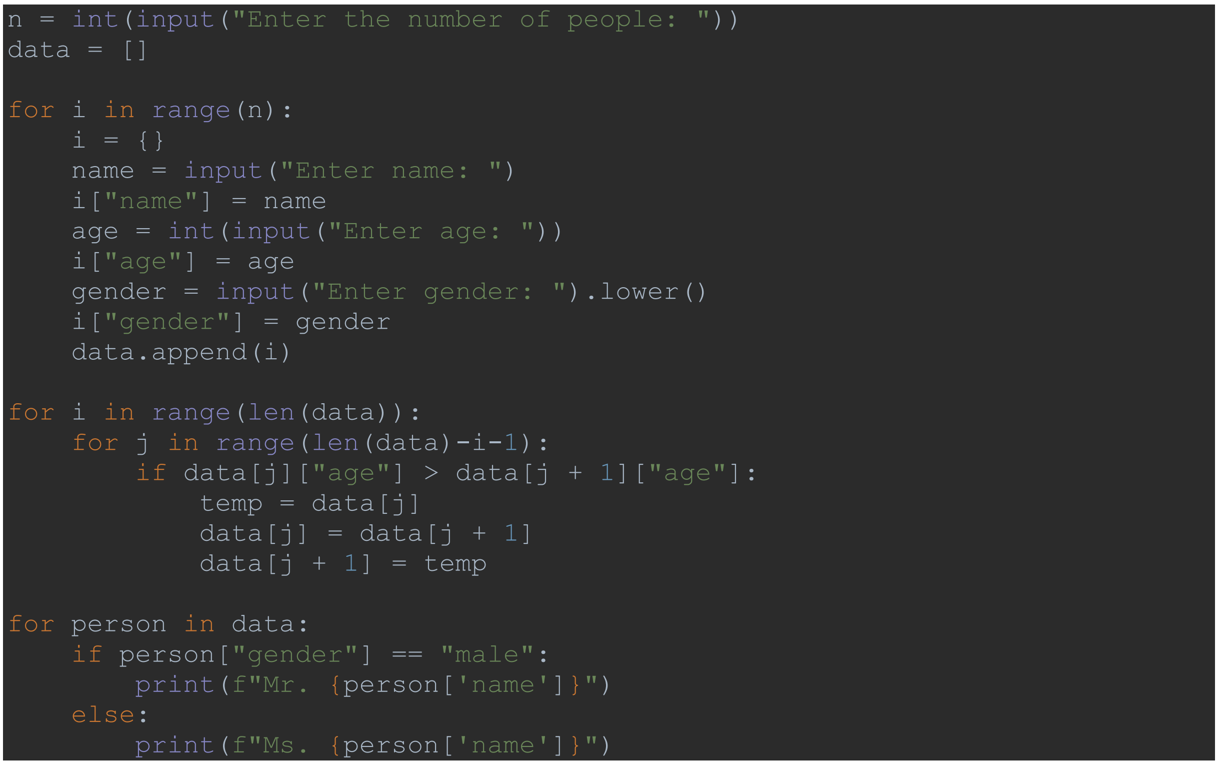
N lines follow each containing the space separated values of the first name, last name, age and sex, respectively.

Constraints

1<=N<=10

Output Format

N names on separate lines in the format described above in ascending order of age.



```
n = int(input("Enter the number of people: "))
data = []

for i in range(n):
    i = {}
    name = input("Enter name: ")
    i["name"] = name
    age = int(input("Enter age: "))
    i["age"] = age
    gender = input("Enter gender: ").lower()
    i["gender"] = gender
    data.append(i)

for i in range(len(data)):
    for j in range(len(data)-i-1):
        if data[j]["age"] > data[j + 1]["age"]:
            temp = data[j]
            data[j] = data[j + 1]
            data[j + 1] = temp

for person in data:
    if person["gender"] == "male":
        print(f"Mr. {person['name']}")
    else:
        print(f"Ms. {person['name']}")
```

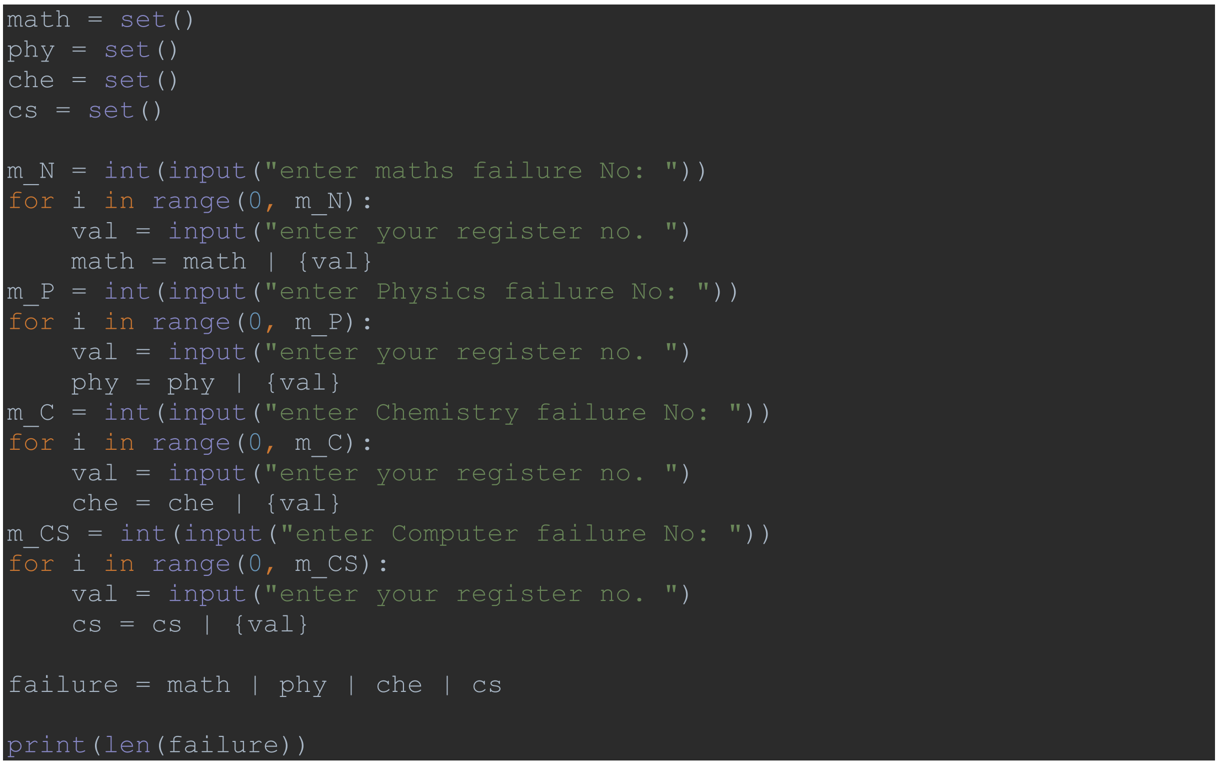
Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX6P5.py"
Enter the number of people: 3
Enter name: Sayan
Enter age: 23
Enter gender: Male
Enter name: Billie
Enter age: 20
Enter gender: Female
Enter name: Leonel
Enter age: 34
Enter gender: Male
Ms. Billie
Mr. Sayan
Mr. Leonel

Process finished with exit code 0
```

Name: SANJAY NAIK
Reg No: 21MCA0017

Exercise 7

Q.1 An University has published the results of the term end examination conducted in April. List of failures in physics, mathematics, chemistry and computer science is available. Write a program to find the number of failures in the examination. This includes the count of failures in one or more subjects. Read the register number of failures in Math, Physics, Chemistry and Computer Science and display the count of failures.



```
math = set()
phy = set()
che = set()
cs = set()

m_N = int(input("enter maths failure No: "))
for i in range(0, m_N):
    val = input("enter your register no. ")
    math = math | {val}
m_P = int(input("enter Physics failure No: "))
for i in range(0, m_P):
    val = input("enter your register no. ")
    phy = phy | {val}
m_C = int(input("enter Chemistry failure No: "))
for i in range(0, m_C):
    val = input("enter your register no. ")
    che = che | {val}
m_CS = int(input("enter Computer failure No: "))
for i in range(0, m_CS):
    val = input("enter your register no. ")
    cs = cs | {val}

failure = math | phy | che | cs

print(len(failure))
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P1.py"
enter maths failure No: 2
enter your register no. 1
enter your register no. 2
enter Physics failure No: 3
enter your register no. 4
enter your register no. 5
enter your register no. 6
enter Chemistry failure No: 4
enter your register no. 7
enter your register no. 8
enter your register no. 9
enter your register no. 10
enter Computer failure No: 1
enter your register no. 69
10

Process finished with exit code 0
```

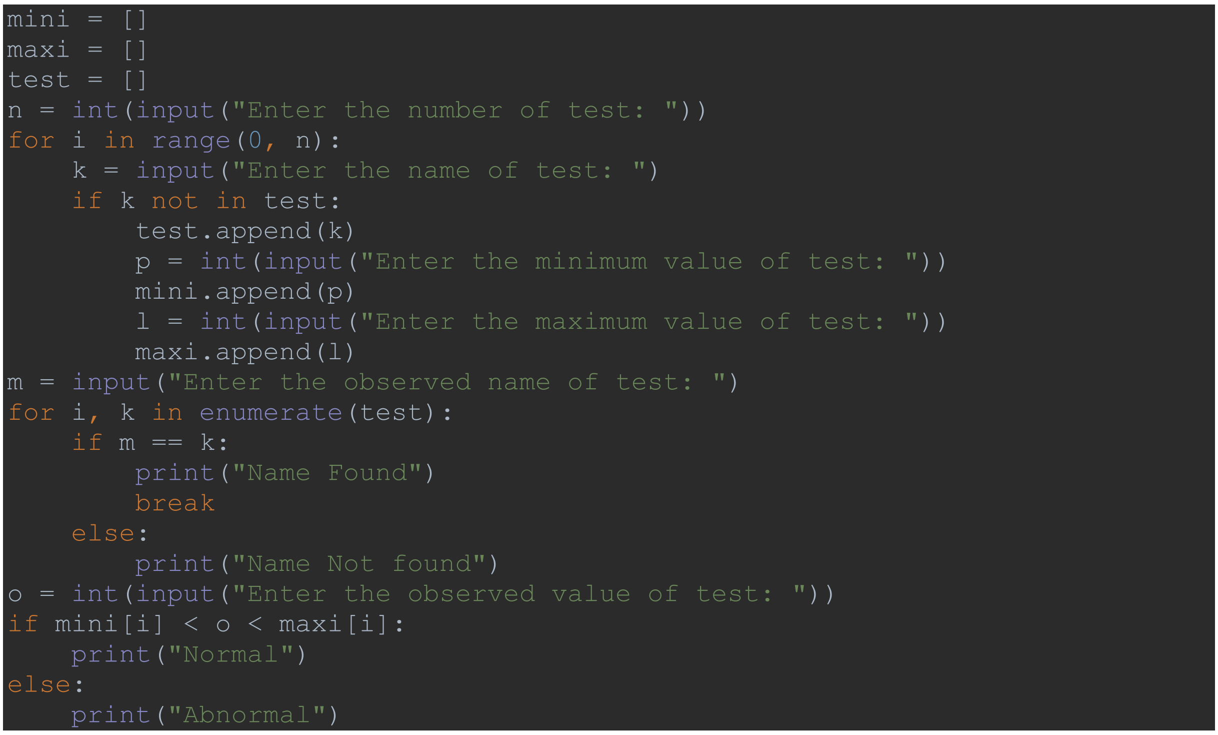
Q.2 A physician wants to check the health status of the patient after performing a few medical tests. Providing the number of tests, get the names of each test, minimum and maximum values corresponding to each one. Further, with the help of the given test name and the observed value of the test of an individual patient, write an algorithm and the subsequent Python program to print if the test result is normal or abnormal. The patient report is normal if the observed value of the test lies in between the maximum and minimum value.

<u>Input Format:</u>

Number of tests
Name of the test
Minimum value of test
Maximum value of test
Name of the observed test
Observed value of the test

<u>Output Format:</u>

Abnormal or Normal



```
mini = []
maxi = []
test = []
n = int(input("Enter the number of test: "))
for i in range(0, n):
    k = input("Enter the name of test: ")
    if k not in test:
        test.append(k)
        p = int(input("Enter the minimum value of test: "))
        mini.append(p)
        l = int(input("Enter the maximum value of test: "))
        maxi.append(l)
m = input("Enter the observed name of test: ")
for i, k in enumerate(test):
    if m == k:
        print("Name Found")
        break
    else:
        print("Name Not found")
o = int(input("Enter the observed value of test: "))
if mini[i] < o < maxi[i]:
    print("Normal")
else:
    print("Abnormal")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P2.py"
Enter the number of test: 2
Enter the name of test: Blood Test
Enter the minimum value of test: 150
Enter the maximum value of test: 200
Enter the name of test: BP
Enter the minimum value of test: 120
Enter the maximum value of test: 269
Enter the observed name of test: Blood Test
Name Found
Enter the observed value of test: 156
Normal

Process finished with exit code 0
```

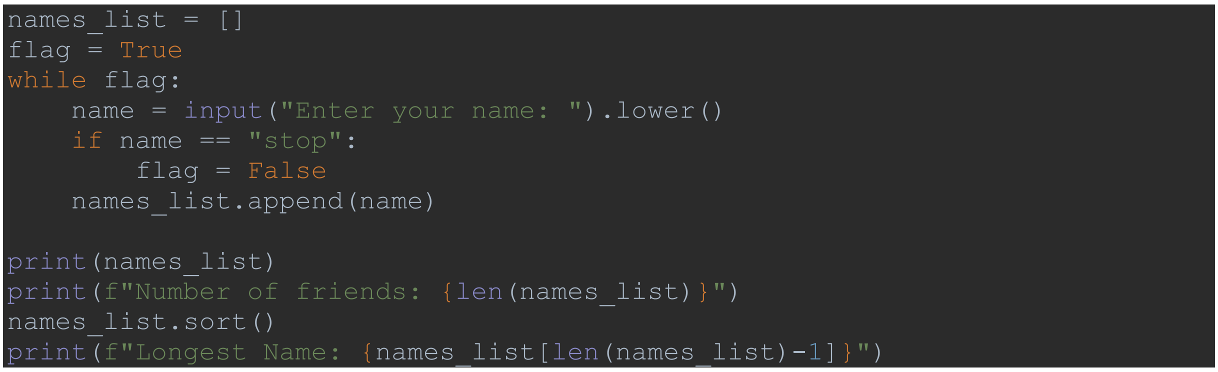
Q.3 Get the names of your friends from the user until user gives ‘STOP’ and store it in a list. Display the friend names in the list format, count the number of friends and find the length of their names. Finally, display friend name with longest name.

<u>Input Format:</u>

Friend names from user until user gives ‘STOP’

<u>Output Format:</u>

List of friend names
Number of friends
List of the length of their names
Friend with longest name



```python
names_list = []
flag = True
while flag:
    name = input("Enter your name: ").lower()
    if name == "stop":
        flag = False
    names_list.append(name)

print(names_list)
print(f"Number of friends: {len(names_list)}")
names_list.sort()
print(f"Longest Name: {names_list[len(names_list)-1]}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EXP3.py"
Enter your name: Sayan
Enter your name: Suman
Enter your name: Sanjay
Enter your name: Leo
Enter your name: Narine
Enter your name: Vaijayantimala
Enter your name: stop
['sayan ', 'suman', 'sanjay', 'leo', 'narine', 'vaijayantimala', 'stop']
Number of friends: 7
Longest Name: vaijayantimala

Process finished with exit code 0
```

Q.4 An anagram is a word or phrase formed by rearranging the letters in another word or phrase. Take two words or phrase from the user and display if they are "anagram" or "not anagram".

Input Format:

First word or phrase
Second word or phrase

Output Format:

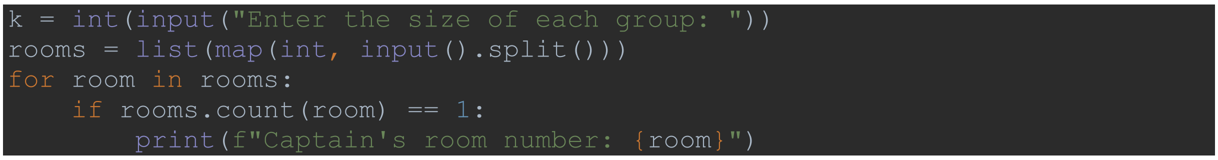
"anagram" or "not anagram"

```
def check(s1, s2):
    if sorted(s1) == sorted(s2):
        print("The strings are anagrams.")
    else:
        print("The strings aren't anagrams.")


s1 = input("Enter first word: ")
s2 = input("Enter second word: ")
check(s1, s2)
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P4.py"
Enter first word: silent
Enter second word: listen
The strings are anagrams.

Process finished with exit code 0
```

Q.5 Mr. Bobby is the manager at the GINGER hotel. The hotel has an infinite amount of rooms. One fine day, a finite number of tourists come to stay at the hotel. The tourists consist of: → A Captain. → An unknown group of families consisting of K members per group where K ≠1 . The Captain was given a separate room, and the rest were given one room per group. Mr. Bobby has an unordered list of randomly arranged room entries. The list consists of the room numbers for all of the tourists. The room numbers will appear K times per group except for the Captain's room.
Mr. Bobby needs you to help him find the Captain's room number. The total number of tourists or the total number of groups of families is not known to you. You only know the value of K and the room number list.
<u>Input Format</u>
The first line consists of an integer, K, the size of each group. The second line contains the unordered elements of the room number list.
<u>Constraints</u>
1<K<1000
<u>Output Format</u>
Output the Captain's room number. Sample Input 5 1 2 3 6 5 4 4 2 5 3 6 1 6 5 3 2 4 1 2 5 1 4 3 6 8 4 3 1 5 6 2 Sample Output 8
<u>Explanation:</u> The list of room numbers contains 31 elements. Since K is 5, there must be 6 groups of families. In the given list, all of the numbers repeat 5 times except for room number 8. Hence, 8 is the Captain's room number.



```
k = int(input("Enter the size of each group: "))
rooms = list(map(int, input().split()))
for room in rooms:
    if rooms.count(room) == 1:
        print(f"Captain's room number: {room}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P5.py"
Enter the size of each group: 5
1 2 3 6 5 4 4 2 5 3 6 1 6 5 3 2 4 1 2 5 1 4 3 6 8 4 3 1 5 6 2
Captain's room number: 8

Process finished with exit code 0
```

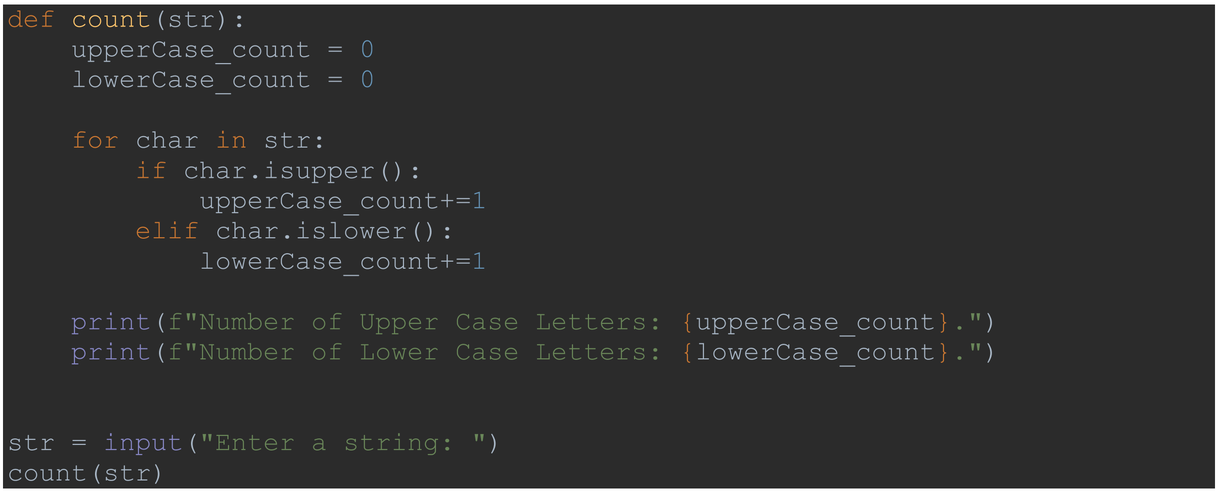
Q.6 Write a Python function that accepts a string and calculate the number of upper-case letters and lower-case letters.

<u>Input Format:</u>

A string

<u>Output Format:</u>

Number of upper-case letters
Number of lower-case letters



```python
def count(str):
    upperCase_count = 0
    lowerCase_count = 0

    for char in str:
        if char.isupper():
            upperCase_count+=1
        elif char.islower():
            lowerCase_count+=1

    print(f"Number of Upper Case Letters: {upperCase_count}.")
    print(f"Number of Lower Case Letters: {lowerCase_count}.")


str = input("Enter a string: ")
count(str)
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P6.py"
Enter a string:
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX7P6.py"
Number of Upper Case Letters: 13.
Number of Lower Case Letters: 51.

Process finished with exit code 0
```

Exercise 8

Q1 Same as Exercise 7 Q4

Q2 Same as Exercise 7 Q6

Q.3 Given a string. The task is to replace each character of the minimized string by a character present at index 'IND' of the original string result in final string. The minimized string is the string obtained by removing all duplicates from the original string keeping the order of elements same. Obtain new string by adding the final string and the original string. Write a function to perform the bubble sort of the characters present in new string. Write a Python program to implement the Scenario.

IND for any index in the minimized string is calculated as:

IND = (square of ascii value of minimized string character) % (length of original string)

Input Format:

String

Output Format:

List of minimized string's characters

New string

Character sorting in a new string

Examples:

Input : geeks

Output :

['g', 'e', 'k', 's']

sesggeeks

['s', 's', 's', 'k', 'g', 'g', 'e', 'e', 'e']

Explanation : minimized string = geks

length of original string = 5

ascii value of g = 103

IND for g = (103*103) % 5 = 4

replacement character for g = s

character 's' present at index 4 of original string

Similarly,

replacement character for e = e

replacement character for k = s

replacement character for s = g

New String=sesg+geeks

```
def minimize(string):
    mstr = " "
    flagchar = [0] * 26
    l = len(string)

    for i in range(0, len(string)):
        ch = string[i]
        if flagchar[ord(ch) - 97] == 0:
            mstr = mstr + ch
            flagchar[ord(ch) - 97] = 1

    return mstr  # minimized string

def replaceMinimizeUtil(string):
    finalStr = ""
    l = len(string)
    minimizedStr = minimize(string)

    for i in range(0, len(minimizedStr)):
        ch = ord(minimizedStr[i])
        index = (ch * ch) % l
        finalStr = finalStr + string[index]

    print("Final string:", finalStr)



string = input("Enter a String: ")
replaceMinimizeUtil(string)
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX8P3.py"
Enter a String: geeks
Final string: ssesg

Process finished with exit code 0
```

Q.4 <u>Radha and Toys</u>

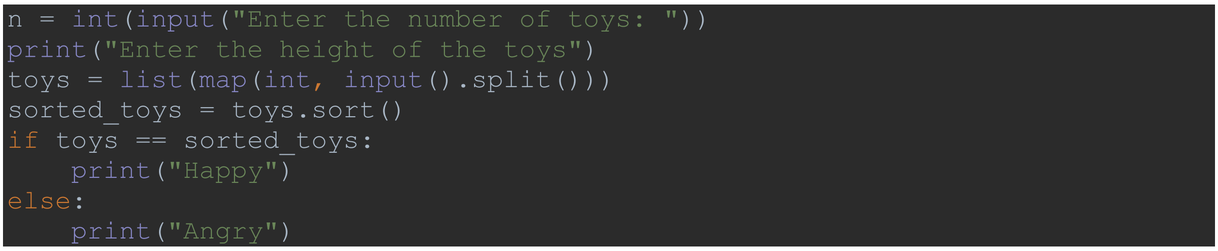
Radha has many toys and she has a weird obsession of arranging all the toys in a line such that their heights are in non-decreasing order. If the order is messed then she gets angry. You know the number of toys and their current arrangement; can you tell if she is angry or not?

<u>Input Format:</u>

First line will contain n the number of toys Monica has. Next line contains n integers, denoting the heights of the toys in line.

<u>Output Format:</u>

If Radha is angry print "Angry" (without quotes) else print "Happy".



```python
n = int(input("Enter the number of toys: "))
print("Enter the height of the toys")
toys = list(map(int, input().split()))
sorted_toys = toys.sort()
if toys == sorted_toys:
    print("Happy")
else:
    print("Angry")
```

Output

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX8P4.py"
Enter the number of toys: 3
Enter the height of the toys
123
Happy

Process finished with exit code 0
```

Q5. The CTS in VIT Vellore wants to create email id to all the first-year students. Assume that they get the current email id of the students, separate the username from it and append the new domain as "vit.ac.in" to create the new vit account. Given a personal mail id of student, design a flowchart and write the Python code to automatically generate the new mail id. If the current mail id is deepakshah@gmail.com then the new mail id to be generated is deepakshah@vit.ac.in

Input Format

Current email id

Output Format

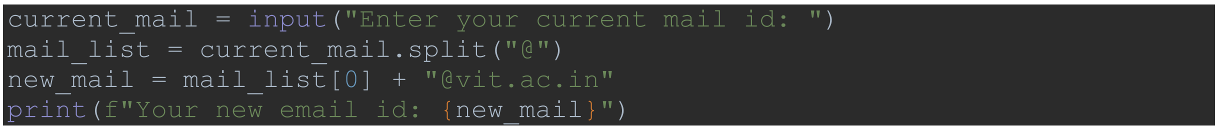
New email id

Example Input/Output 1:

deepakshah@gmail.com

Output:

deepakshah@vit.ac.in



```
current_mail = input("Enter your current mail id: ")
mail_list = current_mail.split("@")
new_mail = mail_list[0] + "@vit.ac.in"
print(f"Your new email id: {new_mail}")
```

Output:

```
C:\Users\login\AppData\Local\Programs\Python\Python39\python.exe
"E:/Python 2022/EX8P5.py"
Enter your current mail id:SANJAYNAIKgmail.com
Your new email id: SANJAYNAIK@vit.ac.in

Process finished with exit code 0
```