```
'''
//===============================================================================
// Name        : 21465_Pract2.py
// Author      : Chaitanya Paraskar
// Roll No.    : 21465
// Aim         : Write a Python program to compute following operations on
String:
                a) To display word with the longest length
                b) To determines the frequency of occurrence of particular
character in the string
                c) To check whether given string is palindrome or not
                d) To display index of first appearance of the substring
                e) To count the occurrences of each word in a given string
                (Do not use string built-in functions)
//===============================================================================

'''


class Str_Op:
    def present(self, v, li):
        for i in range(len(li)):
            if v == li[i]:
                return True
        else:
            return False

    def not_present(self, v, li):
        for i in range(len(li)):
            if v == li[i]:
                return False
        else:
            return True

    def uniqueChars(self, str):
        res = []
        for i in range(0, self.length(str)):
            if self.not_present(str[i], res):
                res.append(str[i])
        return res

    def uniqueWords(self, str):
        res = []
        li = self.Sep_Word(str)
        for i in range(0, self.lengthList(li)):
            if self.not_present(li[i], res):
                res.append(li[i])
        return res

    def Sep_Word(self, str):
        res = []
        word = ""
        str = str + " "
        for c in str:
            if c != " ":
                word = word + c
            else:
                res.append(word)
                word = ""
        return res

    def longest(self):
        str = input("Enter your String : ")

        li = self.Sep_Word(str)

        longest_word = ""
        longest_len = self.length(longest_word)
        current_word = ""
        current_len = self.length(current_word)

        print(li)

        for word in li:
            current_word = word
            current_len = self.length(word)

            if current_len >= longest_len:
                longest_word = current_word
                longest_len = current_len

        print("Longest Word =", longest_word)
        print("Length of longest word =", longest_len)

    def occurance_char(self):
        str = input("Enter your String : ")

        chars = self.uniqueChars(str)
        count = [0] * self.lengthList(chars)

        for i in range(0, self.lengthList(chars)):
            for j in range(0, self.length(str)):
                if str[j] == chars[i]:
                    count[i] = count[i] + 1

        for i in range(0, self.lengthList(chars)):
            print(f"Count of {chars[i]} : {count[i]}")

    def occurance_word(self):
        str = input("Enter your String : ")

        li = self.Sep_Word(str)

        words = self.uniqueWords(str)
        count = [0] * self.lengthList(words)

        for i in range(0, self.lengthList(words)):
            for j in range(0, self.lengthList(li)):
                if li[j] == words[i]:
                    count[i] = count[i] + 1

        for i in range(0, self.lengthList(words)):
            print(f"Count of {words[i]} : {count[i]}")

    def substring(self):
        str = input("Enter your String : ")
        p = input("Enter Sub String to Search : ")

        strl = self.length(str)
        pl = self.length(p)

        found = False
        index = -1

        for i in range(strl - pl + 1):
            match = True
            for j in range(pl):
                if str[i + j] != p[j]:
                    match = False
                    break
                if match:
                    index = i
                    found = True

        if found:
            print(f"{p} found at {index}th index")
        else:
            print(f"Not Found !!")

    def palindrome(self):
        str = input("Enter your String : ")
        isPalindrome = True

        for i in range(0, self.length(str) // 2):
            if not str[i] == str[-1-i]:
                isPalindrome = False
                break
        if isPalindrome:
            print("String is a Palindrome !!")
        else:
            print("String is not a Palindrome !!")

    def length(self, str):
        str = str + "\0"
        i = 0
        while str[i] != "\0":
            i = i + 1
        return i

    def lengthList(self, li):
        li.append("\0")
        i = 0
        while li[i] != "\0":
            i = i + 1
        return i


def main():
    while True:
        obj = Str_Op()

        print("1. Display Longest Word")
        print("2. Display Frequency of occurence of Characters")
        print("3. Check Whether given string is palindrome or not")
        print("4. Display index of 1st appearance of sub string")
        print("5. Display frequency of occurence of Words")
        print("6. Exit")

        ch = int(input("Enter choice"))

        if ch == 1:
            obj.longest()
        elif ch == 2:
            obj.occurance_char()
        elif ch == 3:
            obj.palindrome()
        elif ch == 4:
            obj.substring()
        elif ch == 5:
            obj.occurance_word()
        elif ch == 6:
            break
        else:
            print("Enter Valid Input")


main()


'''
OUTPUT:

$ python pract2.py
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice1
Enter your String : Hello!! I am Chaitanya
['Hello!!', 'I', 'am', 'Chaitanya']
Longest Word = Chaitanya
Length of longest word = 9
1. Display Longest Word
2. Display Frequency of occurence of Characters
```

```
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice2
Enter your String : Heleleleleleo
Count of H : 1
Count of e : 6
Count of l : 5
Count of o : 1
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice3
Enter your String : IammaI
String is a Palindrome !!
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice1
Enter your String : Helloeleleleello
['Helloeleleleello']
Longest Word = Helloeleleleello
Length of longest word = 16
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice3
Enter your String : rhrh
String is not a Palindrome !!
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice4
Enter your String : nandnandan
Enter Sub String to Search : an
an found at 8th index
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
```

```
6. Exit
Enter choice5
Enter your String : Hello I am Chaitanya, I am a Student I am currently admitted
to PICT
Count of Hello : 1
Count of I : 3
Count of am : 3
Count of Chaitanya, : 1
Count of a : 1
Count of Student : 1
Count of currently : 1
Count of admitted : 1
Count of to : 1
Count of PICT : 1
1. Display Longest Word
2. Display Frequency of occurence of Characters
3. Check Whether given string is palindrome or not
4. Display index of 1st appearance of sub string
5. Display frequency of occurence of Words
6. Exit
Enter choice6

'''
```