```python
# 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


def display_longest_word(input_string):

    words = input_string.split()

    longest_word = max(words, key=len)

    print(f"The longest word is: {longest_word}")


def frequency_of_character(input_string, char):

    char_frequency = input_string.count(char)

    print(f"The frequency of '{char}' in the string is: {char_frequency}")


def is_palindrome(input_string):

    reversed_string = input_string[::-1]

    if input_string == reversed_string:

        print("The string is a palindrome.")

    else:

        print("The string is not a palindrome.")


def index_of_substring(input_string, substring):

    index = input_string.find(substring)
```

```python
    if index != -1:
        print(f"The index of the first appearance of '{substring}' is: {index}")
    else:
        print(f"'{substring}' not found in the string.")


def count_word_occurrences(input_string):
    words = input_string.split()
    word_count = {}

    for word in words:
        word_count[word] = word_count.get(word, 0) + 1

    print("Word occurrences:")
    for word, count in word_count.items():
        print(f"{word}: {count} times")


def main():
    input_string = input("Enter a string: ")

    display_longest_word(input_string)

    char_to_find = input("Enter a character to find its frequency: ")
    frequency_of_character(input_string, char_to_find)

    is_palindrome(input_string)
```

```python
    substring_to_find = input("Enter a substring to find its index: ")

    index_of_substring(input_string, substring_to_find)


    count_word_occurrences(input_string)


if __name__ == "__main__":
    main()
```