Assignment 7

August 18, 2023

1 Assignment 7

1.0.1 Benjamin Michaels

1.0.2 8/18/23

1.1 Question 1

A palindrome is a word, phrase, or sequence that is the same spelled forward as it is backwards. Write a function using a for-loop to determine if a string is a palindrome. Your function should only have one argument.

[1]:

```
# your code here

word = input("Enter a word: ")c
word_lower =word.lower()
len_word = len(word)
i = 0
for i in range(len_word):
    if word_lower[i] == word_lower[-1-i]:
            print(word, "is a palindrome.")
            break
    else:
        print(word, "is not a palindrome.")
        break
```

```
Enter a word: racecar
racecar is a palindrome.
```

1.2 Question 2

Write a function using a while-loop to determine if a string is a palindrome. Your function should only have one argument.

[2]:

```
# your code here

word = input("Enter a word: ")
word_lower =word.lower()
len_word = len(word)
revWord = ""
```

```python
i=len_word-1

while i >= 0:
    revWord = revWord + word_lower[i]
    i=i-1
if(word_lower == revWord):
    print(word, "is a palindrome.")
else:
    print(word, "is not a palindrome.")
```

```
Enter a word: Civic
Civic is a palindrome.
```

1.3 Question 3

Two Sum - Write a function named two_sum() Given a vector of integers nums and an integer target, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.Use defaultdict and hash maps/tables to complete this problem.

Example 1: Input: `nums = [2,7,11,15]`, target = 9 Output: `[0,1]` Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2: Input: `nums = [3,2,4], target = 6 Output: [1,2]`

Example 3: Input: `nums = [3,3], target = 6 Output: [0,1]`

Constraints: `2 <= nums.length <= 104 -109 <= nums[i] <= 109 -109 <= target <= 109` Only one valid answer exists.

[3]:

```python
# your code here

def two_sum(nums_vector,target):
    for i in range(0,len(nums_vector)):
        for j in range(i+1,len(nums_vector)):
            if(int(nums_vector[i])+int(nums_vector[j])==target):
                return(i,j)

nums_vector = [3,2,4]
target = 6
two_sum(nums_vector,target)
```

[3]:

```
(1, 2)
```

1.4 Question 4

How is a negative index used in Python? Show an example

[7]:

```python
# your code here
```

```
friends = ["Ben", "Alex", "Cole", "Bonnie"]
friends[-1]
```

1.5 Question 5

Check if two given strings are isomorphic to each other. Two strings str1 and str2 are called isomorphic if there is a one-to-one mapping possible for every character of str1 to every character of str2. And all occurrences of every character in ‘str1’ map to the same character in ‘str2’.

`Input:  str1 = "aab", str2 = "xxy"`

`Output: True`

`'a' is mapped to 'x' and 'b' is mapped to 'y'.`

`Input:  str1 = "aab", str2 = "xyz"`

`Output: False`

`One occurrence of 'a' in str1 has 'x' in str2 and other occurrence of 'a' has 'y'.`

A Simple Solution is to consider every character of ‘str1’ and check if all occurrences of it map to the same character in ‘str2’. The time complexity of this solution is O(n*n).

An Efficient Solution can solve this problem in O(n) time. The idea is to create an array to store mappings of processed characters.

[ ]:

```
# your code here
def two_sum(str1,str2):
    if len(str1) != len(str2):
        return False
    else:
        x = {}
        y={}
        for i in range(len(str1)):
            char1 = str1[i]
            char2 = str2[i]
            if char1 not in x:
                x[char1]=char2
            if char2 not in y:
                y[char2]=char1
            if x[char1] != char2 or y[char2] != char1:
                return False
            return True

string1 = input("Input the first string: ")
string2 = input("Input the second string: ")
print(two_sum(string1,string2))
```

[ ]: