Assignment -3

Program 1

```
# Assignment 3
# 1. Python function that takes a list of words and return the  longest word and
 the length of the longest one.

'''
Initially, wordLength = 1
Iteration 1:
word = "Apple"
len(word) = 5 > wordLength(1)
Therefore, position = 0, wordLength = 5
.
.
.
Iteration 6:
word = "WaterMelon"
len(word) = 10 > wordLength
Therefore, position = 5, wordLength = 10
Iteration 7:
word = "Pomegranate"
len(word) = 11 > wordLength(10)
Therefore, position = 6, wordLength = 11
.
.
.
Iteration 10:
word = "Orange"
len(word) = 6 > wordLength(11) -> False
Therefore, position = 6, wordLength = 11

'''
def longWord(fruits):
  wordLength = 1
  for word in fruits:
    if len(word) >= wordLength:
      position = fruits.index(word)
      wordLength = len(word)
  return position

words = ["Apple", "Mango", "Banana", "Pineapple", "MuskMelon", "WaterMelon", "Po
megranate", "Lychee", "StrawBerry", "Orange"]
# calling function longWord() and passing the words list as actual parameter.
pos = longWord(words)
print("The longest word is ",words[pos], "with length", len(words[pos]))
```

Output

```
The longest word is  Pomegranate with length 11
```

Program 2

```
# 2. Python function to remove the nth index character from a nonempty string.
'''
String is immutable in Python.
Removing a character at nth index will require to form a new string.
This can be done by removing the nth index character using string slicing.
Here, original string = "This is a new journey."
Characters:T h i s   i s   a    n  e  w     j  o  u  r  n  e  y  .
Index:     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

During the function call removeChar(pos, word),
len(word) ->22 > 0 . The string is not empty
new_word = word[:pos]+word[pos+1:] -
> character at index pos will not be added to the new word
new_word = word[:11]+word[12:] -
> character at index 11 will not be added to the new word

'''
def removeChar(pos, word):
  if len(word)>0:
    new_word = word[:pos]+word[pos+1:]
  return new_word

s = "This is a new journey."
p = 11
new_s = removeChar(p,s)
print("Original string:", s)
print("After removing character",s[p]," at index:",p,", new string:", new_s)
```

Output

```
Original string: This is a new journey.
After removing character e  at index: 11 , new string: This is a nw journey.
```

Program 3

```
# 3. Python function to get the last part of a string before a specified charact
er.
'''
To get the last part of a string before a specified character we use a function
called rpartion()
str.rpartition(sep)
Split the string at the last occurrence of sep, and return a 3-
tuple containing the part before the separator,
the separator itself, and the part after the separator. If the separator is not
found,
 return a 3-tuple containing two empty strings, followed by the string itself.

Original string: Welcome to python. Python is used for Data Science; I am attend
ing a course for Data Science at JNTUH SCDE!
Character = '.'
After o_string.rpartition(ch)
new_string = ('Welcome to python', '.', ' Python is used for Data Science; I am
attending a course for Data Science at JNTUH SCDE!')
new_string[0] = Welcome to python
'''
def lastPart(o_string, ch):
  new_string = o_string.rpartition(ch)
  return new_string

s = "Welcome to python. Python is used for Data Science; I am attending a course
 for Data Science at JNTUH SCDE!"
c = '.'
new_s = lastPart(s, c)
print("Original string:", s)
print("Character:", c)
print("New String:", new_s[0])
```

Output

```
Original string: Welcome to python. Python is used for Data Science; I am
attending a course for Data Science at JNTUH SCDE!
Character: .
New String: Welcome to python
```

Program 4

```
# 4. Python function to sort a string lexicographically.

  def lexi(names):
    new_names = sorted(names)
    return new_names
  nl = ["John", "Mary", "Ashutosh", "Devin", "Suchit", "Sai Priya", "Ankita"]
  n = lexi(nl)
  print(n)
```

Output

```
['Ankita', 'Ashutosh', 'Devin', 'John', 'Mary', 'Sai Priya', 'Suchit']
```

Program 5

```
# 5. Python function to remove spaces from a given string.
def removeSpaces(s):
  new_s = s.replace(" ","")
  return new_s
# generate any random sentence : https://randomwordgenerator.com/sentence.php
s = "The Great Dane looked more like a horse than a dog."
print(removeSpaces(s))
```

Output

```
TheGreatDanelookedmorelikeahorsethanadog.
```