```
#1
#Dictionary of charecters and its index pair
#INPUT - string = 'Malayalam'
#OUTPUT - defaultdict(<class 'list'>, {'M': [0], 'a': [1, 3, 5, 7], 'l': [2, 6], 'y': [4], 'm': [8, 8]})

string = 'Malayalam'
from collections import defaultdict

d = defaultdict(list)

for index, char in enumerate(string):
   d[char] += [index]
d[char].append(index)

print(d)

#defaultdict(<class 'list'>, {'M': [0], 'a': [1, 3, 5, 7], 'l': [2, 6], 'y': [4], 'm': [8, 8]})

#*****************************************************************************************
*****************************************

#2
#Write a program to create a dictionary of 1st charecter and the word starting with that first charecter pair in the given sentence

string = 'komal komal A'
words = string.split()
d = {}

for word in words:
```

```
    if word[0] not in d.keys():
        d[word[0]] = []
        d[word[0]].append(word)
    if word not in d[word[0]]:
        d[word[0]].append(word)
print(d)
{'k': ['komal'], 'A': ['A']}



#*****************************************************************************************
*****************************************
#3.
#To find the len of iterable without using any inbuilt function
string = 'Komal'
count = 0
for char in string:
    count += 1
print(count)

#4
#WAP reverse a string without using any inbuilt method
res = string[::-1]
print(res)

#5
#WAP to replace one string with another
stg = 'Hello World'
old = 'World'
new = 'Universe'
```

```
res = ''

words = stg.split()
for word in words:
   if word == old:
      res += new + ' '
   else:
      res += word +  ' '
print(res)

#6
#WAP to convert string into list and vice versa
string = 'Hello World'
list = string.split()
print(list)
res = ' '.join(list)
print(res)

#7
#WAP to convert string into comma separated value
string = 'Hello World'
res = ','.join(list)
print(res)



#8
#check_string
string = 'Hello World'
res = isinstance(string, str)
```

```
print(res)


#9
#Write a Python program to swap cases of a given string.
def swap(char):
    if char.islower():
        return char.upper()
    elif char.isupper():
        return char.lower()
print(swap('a'))


def swap(char):
    if ord('a')<= ord(char)<= ord('z'):
        return chr(ord(char) - 32)
    elif ord('A')<= ord(char)<= ord('Z'):
        return chr(ord(char) +32)
print(swap('B'))


#10
#palindrome
string = 'MOM'


def pal(word):
    res = ''
    for word in string:
        res = word + res
    if res == string:
        return 'Palindrome'
    return 'Not palindrome'
```

```
print(pal(string))


#11
#replace vowels with *
string = 'Komal'
def isvowels(name):
    res = ''
    for char in string:
        if char[0] in 'aeiou':
            res += '*'
        else:
            res += char
    return res
print(isvowels(string))

#K*m*l


#12
#replace repeated charecters with -
string = 'Komal Komal A'
def rep(char):
    res = ''
    for char in string:
        if string.count(char) > 1:
            res += '-'
        else:
            res += char
    return res
print(rep(string))
```

```
#13
#Longest word
sentence = 'python is a good programming language'
words= sentence.split()
longest = ""
for word in words:
    if len(word) > len(longest):
        longest = word
print(longest)

#13
#non repeated longest word
sentence = 'python is a good programming language and programming is fun'
words = sentence.split()
longest = ""
for word in words:
    if len(word) > len(longest):
        if sentence.count(word) == 1:
            longest = word
print(longest) #language


#14
#create a dictionary with word with its length pair python program
#{'python': 6, 'is': 2, 'a': 1, 'good': 4, 'programming': 11, 'language': 8} word with its length
sentence = 'python is a good programming language'
words = sentence.split()
d = {word: len(word) for word in words}
```

```
print(d) #{'python': 6, 'is': 2, 'a': 1, 'good': 4, 'programming': 11, 'language': 8}


#15
#longest word and its length
sentence = 'python is a good programming language'
words = sentence.split()
d = {word: len(word) for word in words}
print(d)


res = sorted(d.items())
print(res) #[('a', 1), ('good', 4), ('is', 2), ('language', 8), ('programming', 11), ('python', 6)]


print(res[-2]) #('programming', 11)


#16
#sorted based on len
largest_repeat_word = sorted(d.items(), key= lambda item: item[-1] )
print(largest_repeat_word)


#17
#Charecter count
sentence = 'python is a good programming language'
d={}
for char in sentence:
   if char not in d:
      d[char] = 1
   else:
      d[char] = d[char] + 1
print(d)
```

```
#18
#counting spaces
d={}
for char in sentence:
    if char not in d and char ==  ' ':
        d[char] = sentence.count(char)
print(d)#{' ': 5}
```