Name : Lakhan kumawat

Roll No : 1906055 branch

: CSE-1

Course : CSL5401

Lab 2 Assignment

1. WAP in python to create three different strings,catenate all three strings and print.

```
# -*- coding: utf-8 -*-

import string
import random # define the random module
S = 6  # number of characters in the string.
result=''
for i in range(3):
    ran = ''.join(random.choices(string.ascii_lowercase , k = S))
    print("The randomly generated string is : " + str(ran)) # print the str
ing
    result+=ran
print("Final Catenated String is :"+result)
```

```
>>> %Run main.py
 The randomly generated string is : pvywwa
 The randomly generated string is : krakhg
 The randomly generated string is : aimvnq
 Final Catenated String is :pvywwakrakhgaimvnq
```

2. WAP demonstrating operations on dictionaries .

```
dict_new={}

import operator

for i in range(3):
    v=input("Enter value :")
    dict_new[i]=v
```

```
print("Dictionary Created...")
print(dict_new)
new_value=input("Enter a value to change : ")
dict_new[0]=new_value
print(dict_new)
dict_new.pop(1)
print("After removing an element ", dict_new)
print("length ",len(dict_new))

copied_dic=dict_new.copy()
print("Copied dictionary : ",copied_dic)

print("Sorting basis of values : ", sorted(dict_new.items(),key=operator.it
emgetter(1)))
```

```
Python 3.7.9 (bundled)
>>> %Run main.py
 Enter value :value
 Enter value :user
 Enter value :nul
 Dictionary Created...
 {0: 'value', 1: 'user', 2: 'nul'}
 Enter a value to change : new
 {0: 'new', 1: 'user', 2: 'nul'}
 After removing an element  {0: 'new', 2: 'nul'}
 length  2
 Copied dictionary :  {0: 'new', 2: 'nul'}
 Sorting basis of values :  [(0, 'new'), (2, 'nul')]
>>>
```

3. WAP to that prints all the unique words of a file in alphabetical order .

```
text_file = open('abc.txt', 'r')
text = text_file.read()

#cleaning
text = text.lower()
words = text.split()
words = [word.strip('.,!;()[]') for word in words]
words = [word.replace("'s", '') for word in words]
```

```python
#finding unique
unique = []
for word in words:
    if word not in unique:
        unique.append(word)

#sort
unique.sort()


#print

for word in unique:
    print(word)
```

```
>>> %Run main.py
 a
 and
 code:
 content
 description:
 file
 from
 in
 involves
 it
 printing
 program
 python
```