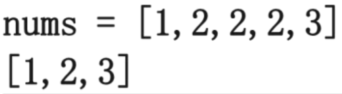
NAME : W.VIVIYAN RICHARDS
DEP.NO :205229133
SUBJECT : PYTHON
ASSIGNMENT :PSPR LAB ASSIGNMENT-9

Question7. Develop a function remove_adjacent(). Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element. You may create a new list or the modify the passed in list.

Source code:

```
def remove_adjacent(nums):
result = [] for
num in nums:
if len(result) == 0 or num != result[-1]:
result.append(num) return result
```

Output:



```
nums = [1, 2, 2, 2, 3]
[1, 2, 3]
```

Question8.Write a function verbing(). given a string, if it is at least 3, add 'ing' to its end. Unless it already ends in 'ing',in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string. So 'hail' yieldss: 'hailing;''swimming':

Source code:

```
def verbing(s):
   length = len(s)

   if length > 2: if s[-
      3:] == 'ing':
         s += 'ly'
      else:
         s += 'ing'

   return s
```

Output:

```
>>verbing('hail')
'hailing'
>>verbing('heal')
'healing'
```

Question9. Develop a function not_bad(). Given a string, find the first apperance of the substring 'not' and 'bad'follows the 'not',replace the whole 'not'…'bad' substring with 'good'.

Return the resulting string. So 'This dinner is not that b ad!' yields: This dinner is good!

Source code:

```
def not_bad(s):
        snot = s.find('not') sbad = s.find('bad') if
        sbad>snot: s =
        s.replace(s[snot:(sbad+3)],'good')
        return s
```

Output:

```
>>not_bad("This dinner is not not that bad!") 'This
dinner is good!'

>>not_bad("This cricket match is not not that bad!")
'This cricket match is good!'
```

LAB6.PYTHON FILE PROCESSING Question1.

Write a program for Password Management Sy

stem

Source code:

```
def register():
                 username = input("please input the first 2 letters of your first
  name and your birth year") password = input("please input your
        desired password") file = open("loginfile.txt","a")
        file = write(username)
        file.write("") file.write(password)
        file.write("\n") file.close() if
        login(): print("you are logged
        in....")
        else: print("you aren't logged in!")
def login():
        username = input("please Enter your Name:")
        password = input("please Enter your password:") for
        line in open("loginfile.txt","r").readlines():
               login_info = lint.split()
                      if username == login_info[0] and password == login_info
[1]:
                      print("correct credentials!") return
                      True
               print("Incorrect credentials") return
               False
```

Output:

```
please input the first 2 letters of your first name and
 your birth year: ha1999 please input
your desired password: ds112 You are
logged in…. please enter your name:
sudhan please enter your password: fd34
Incorrect credentials.
```

Question7. Develop a function **remove_adjacent()**. Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element. You may create a new list or modify the passed in list.

Test Cases:

1. Input: [1, 2, 2, 3] and output: [1, 2, 3]
2. Input: [2, 2, 3, 3, 3] and output: [2, 3]
3. Input: [ ]. Output: [ ].
4. Input: [2,5,5,6,6,7]
   Output: [2,5,6,7]
5. Input: [6,7,7,8,9,9]
   Output: [6,7,8,9]

```
def remove_adjacent(nums):
        result = []
    for num in nums:
        if. len(result)==0.or num != result[-1]:
            result.append(num)
    return result.

nums = [1,2,2,3]
remove_adjacent (nums)

nums= [2,2,3,3]
remove_adjacent (nums)

nums=[2,2,3 3]
remove_adjacent (nums)

nums = []
remove_adjacent(nums)

nums = [6,7,7,8,9,9]
remove_adjacent (nums)
```

Question8. Write a function **verbing()**. Given a string, if its length is at least 3, add 'ing' to its end Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string. So 'hail' yields: hailing; 'swimming' yields: swimmingly; 'do' yields: do.

Source code:

```
    def verbing(s):
        length = len(s)
    if length >3:
       if s[-3:]=='ing':
            s+='ly'
     else:
        s+='ing'
      return s.
```

Question9. Develop a function **not_bad()**. Given a string, find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the 'not', replace the whole 'not'...'bad' substring with 'good'.
Return the resulting string. So 'This dinner is not that bad!' yields: This dinner is good!

```
def not_bad(s):
     snot = s.find('not')
     sbad = s.find('bad')
     if sbad>snot:
       s = s.replace(s[snot:(sbad+3)],'good')
   return s.
```

Problem Solving Using Python and R Lab

Lab6. Python File Processing

Question1. Write a program for *Password Management System*

- File creation: Ask user to enter N user names and their passwords. Store usernames and passwords into a file named **"loginfile.txt"**. Store each user and password in one line.
- File Processing: Write a program that opens your "security.txt" file and reads usernames and passwords from it. Store user names in one list and passwords in another lists.
- Querying: ask user to enter user name and password for verification. If they match the values stored in the lists, print a message "Login Successful". Otherwise print a message "Login Failed, try again".

source Code:

```
def regis():
    username = input ("please input the first 2 letters of yourfirst
    name and your birth year")
      password = input("please input your desired password")
        file =open("loginfile.txt", "a")
        file = Write (username)
         file.write(" ")
          file.Write(password)
          file.writ ("\n")
           file.close()
           if login():
              print("you are logged in....")
     else:
        print (" you aren't logged in!")
    def login():
           Username = input(" please Enter your Name :")
            password = input ("please Enter your password:")
          for line in open("loginfile.txt", "r").readlines():
           login_info= lint.split()
 if username == login_info[0] and password== login_info.
   [1]:     Print ("Correct Credentials!")
             return True
           Print (" Incorrect Credentials")
           return False.
```