ENGR 101 – Introduction to Programming

Study Questions – Week 8

1. Given the following function calls and outputs, write the function *word_hist*.

```
#function calls
word_hist("the")
word_hist("ENGR211")
word_hist(213090080)
word_hist(9.12613)

#output
the: ***
ENGR211: *******
213090080: *********
9.12613: *******
```

2. Define a function ***unique_elem*** takes two strings str_1 and str_2 as inputs and prints only the elements that occur in str_1 and not in str_2. **Note:** You are NOT allowed to use the **in** operator.

   >> unique_elem(“hello”, “hola”)

   e

   >> unique_elem(“alyo”, “efendim”)

   a

   l

   y

   o

3. Define a function ***string_reverse*** that takes a string as input and returns a reversed version of it.

   >> print string_reverse(“sehir”)

   rihes

4. Define a function ***repeat_letters*** that takes a string and a number n as its two inputs and prints the string with every letter repeated n times.
   >>print repeat_letters(“bla”, 2)
   “bbllaa”

>>print repeat_letters(“sehir”, 3)
“ssseeehhhiiirrr”

5. Define a boolean function (returns True or False only) ***check_pass*** that asks user to enter a password string and checks for any repeated elements. The function returns True if all the elements are unique (none repeated), and False otherwise. **Note:** You are NOT allowed to use **in** operator.

6. Define a function ***random_pass_gen*** that generates a random password of length n. All the elements of the password generated are lowercase letters from English alphabets.

   Hint: Use random.randint function from the random module. Also, you can define and use alphabets = “abcdefghijklmnopqrstuvwxyz”.

7. Define boolean function (returns True or False only) that takes two strings str1 and str2 as inputs and returns the longer of the two strings.

8. Define a boolean function (returns True or False only) that takes two strings str1 and str2 as inputs and checks if they are the same string or not. **Note:** You are NOT allowed to use **in** operator.

9. Write a function, which takes a string as a parameter and returns the length of that string. Then calling that function check if the length of the string is bigger than 5 it should print out “the string’s length is bigger then 5” otherwise it should print “the string’s length is smaller or equal to 5”. Note: You should not use len function.

10. Write a function, which takes as parameters a string and a number and if the length of the string is equal to the number it should return True otherwise, it should return False.

11. Write a function which takes three arguments (string1,char1,char2), a string and two letters(char1 and char2). If the first letter(char1) is in the string1 change it to char2 and print the string modified. If it is not then it should just print the string as it is.

12. Run the following code and tell the output. What would happen if we change return statement to a print statement.

```
def word(word1):
    a=0
    while True:
        if len(word1)< a:
            return a
        a+=1
    print word("welcome")
```

13. Write a function which takes two arguments a word and a letter. The function should go through the word and when it encounter the letter it should return the part of the word after the letter including the letter given as an argument.

    For cut_string("words","r") it should print rds

14. Write a function which takes as input a sentence and a letter. It should count how many times the letter occurs in it.

15. Write a function which take a string as input and prints each letter of the string n times (n being the place of the letter in that string)

    >>repeat_letter("what")

    w

    hh

    aaa

    tttt

16. Write a function which takes three strings as arguments and returns the string which has the longest length. Write the conditions using logical and operator.

17. Write a function which takes as argument a word and a letter. It will return the index of the second occurrence of the letter in the given word.

    >> find_index(“welcomehome”,”e”)

    6

18. Write a function that removes the n^{th} index character from the non-empty string.

    >>remove_ind(“hello”,2)

    helo

19. Write a function to remove the characters which have odd index values of a given string.

    >>remove_odds(“flower”)

    foe

20. Write a function that change a given string to a new string where the first and last chars have been exchanged.

    >>change_first_last(“sehir”)

    rehis