Lab Report Of

COMPILER DESIGN LAB

Course code: CSE332

Submitted To

Aniruddha Rakshit

Department of CSE

Daffodil International University

Submitted By

Ruhul Amin Parvez

ID: 173-15-10419

Sec: O-13

Date Of Submission: 1512/20

LAB FINAL REPORT ANSWER

Lab Class – 1

Task-1 :- Program to count length of a string.

Code:

"""

Created on Sat Dec 12 20:06:24 2020

@author: RUHUL AMIN PARVEZ

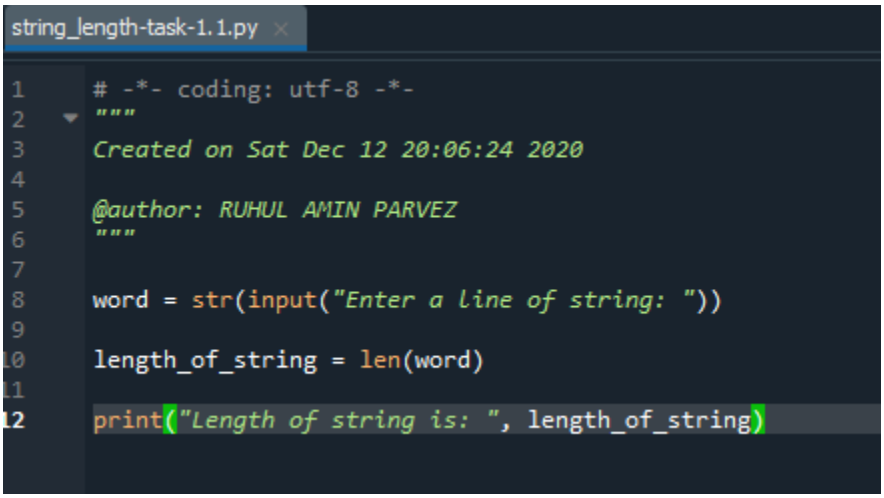
"""

word = str(input("Enter a line of string: "))

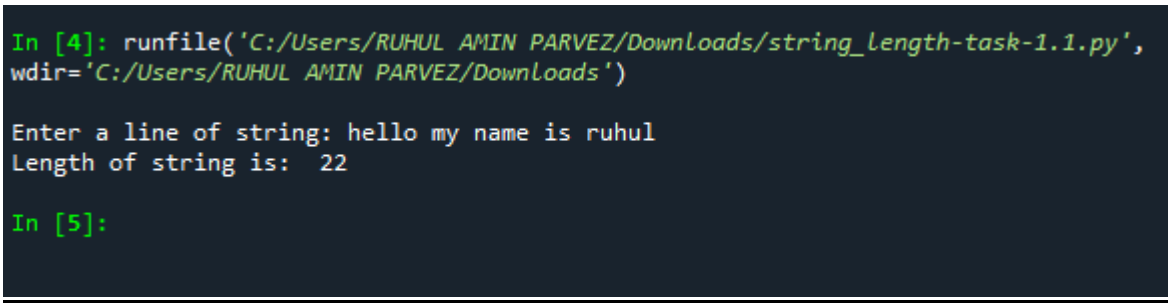
length_of_string = len(word)

print("Length of string is: ", length_of_string)

CODE SCREENSHOT



RUN TEST SCREENSHOT



Task-2 :- Program to search a specific character in a String and if found then print its position.

Code:

```
"""

Created on Sat Dec 12 20:28:45 2020

@author: RUHUL AMIN PARVEZ

"""

ini_string = str(input("Enter a line of string: "))

c = str(input("Enter character to find: "))


res = None

for i in range(0, len(ini_string)):

        if ini_string[i] == c:

                res = i + 1

                break

if res == None:

        print ("No such charater available in string")

else:

        print ("Character {} is present at {}".format(c, str(res)))
```

CODE SCREENSHOT

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 12 20:28:45 2020

@author: RUHUL AMIN PARVEZ
"""


ini_string = str(input("Enter a line of string: "))

# Character to find
c = str(input("Enter character to find: "))



res = None
for i in range(0, len(ini_string)):
    if ini_string[i] == c:
        res = i + 1
        break

if res == None:
    print ("No such charater available in string")
else:
    print ("Character {} is present at {}".format(c, str(res)))

```

RUN TEST SCREENSHOT

```
In [10]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/string_position-
task-1.2.py', wdir='C:/Users/RUHUL AMIN PARVEZ/Downloads')

Enter a line of string: ruhul amin parvez

Enter character to find: p
Character p is present at 12

In [11]:
```

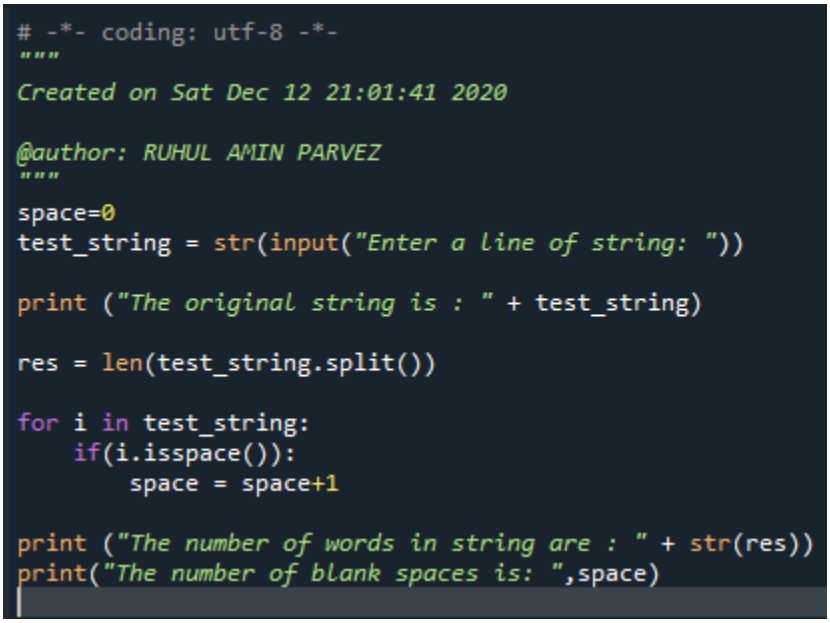
Lab Class – 2

Task-2 :- Program to count words in a sentence. (Condition: count the words with space)

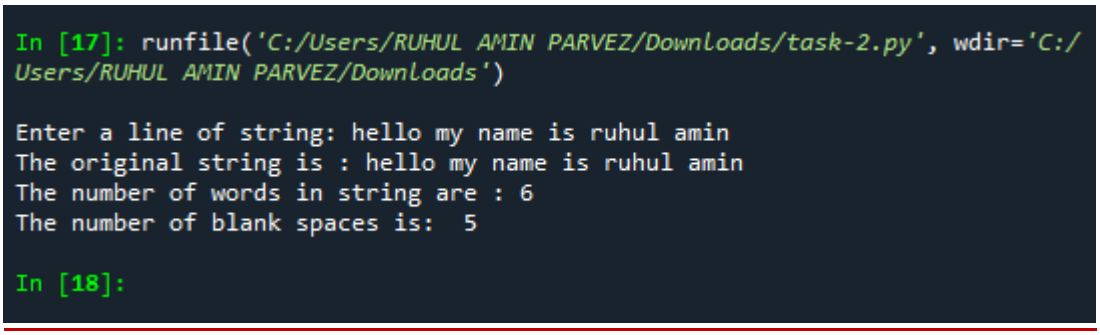
Code:

```
"""
Created on Sat Dec 12 21:01:41 2020
@author: RUHUL AMIN PARVEZ
"""
space=0
test_string = str(input("Enter a line of string: "))
print ("The original string is : " + test_string)

res = len(test_string.split())

for i in test_string:
   if(i.isspace()):
      space = space+1

print ("The number of words in string are : " + str(res))
print("The number of blank spaces is: ",space)
```

CODE SCREENSHOT



RUN TEST SCREENSHOT



Lab Class – 3

Task-1 :- Program to detect single line, multiple line and no comment in a sentence. (Condition: sentence /*sentence*/)

Code:

```
"""

Created on Sat Dec 12 23:57:00 2020

@author: RUHUL AMIN PARVEZ

"""

comment_test = str(input("Enter a line of string: "))


check_one = comment_test[0]

check_two = comment_test[1]


check_last_one = comment_test[-1]

check_last_two = comment_test[-2]


if check_one == "/" and check_two == "/":

    print("It's a single line comment")
```

elif check_one == "/" and check_two == "*" and check_last_two == "*" and check_last_one == "/":

    print("It's a multipleline comment")

else:

    print("It's not a comment")

CODE SCREENSHOT

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 12 23:57:00 2020

@author: RUHUL AMIN PARVEZ
"""

comment_test = str(input("Enter a line of string: "))


check_one = comment_test[0]
check_two = comment_test[1]

check_last_one = comment_test[-1]
check_last_two = comment_test[-2]

if check_one == "/" and check_two == "/":
    print("It's a single line comment")

elif check_one == "/" and check_two == "*"  and check_last_two == "*" and check_last_one == "/":
    print("It's a multipleline comment")

else:
    print("It's not a comment")
```

RUN TEST SCREENSHOT

Case-1

```
In [66]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: multiline /* comment */
It's not a comment
```

Case-2

```
In [67]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: //single line
It's a single line comment
```

Case-3

```
In [68]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: /this is not comment
It's not a comment
```

Case-4

```
In [70]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: single line /comment
It's not a comment
```

Case-5

```
In [71]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: /* multiline comment */
It's a multipleline comment
```

Case-6

```
In [72]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: /* multiline comment
It's not a comment
```

Case-7

```
In [74]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
comment_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this is normal line
It's not a comment
```

Lab Class – 4

Task-1 :- Program to search for a special character in a sentence and if found then print not sentence.

Code:

```
"""

Created on Sun Dec 13 01:14:42 2020

@author: RUHUL AMIN PARVEZ

"""

import re


def run(string):

    check = re.compile('[@_!#$%^&*()<>?/\|}{~:]')


    if(check.search(string) == None):

        print("String is accepted")

    else:

        print("String is not accepted.")


if __name__ == '__main__' :


    string = str(input("Enter a line of string: "))

    run(string)
```

CODE SCREENSHOT

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 13 01:14:42 2020

@author: RUHUL AMIN PARVEZ
"""
import re

def run(string):

    check = re.compile('[@_!#$%^&*()<>?/\|}{~:]')


    if(check.search(string) == None):
        print("String is accepted")

    else:
        print("String is not accepted.")

if __name__ == '__main__' :


    string = str(input("Enter a line of string: "))
    run(string)
```

RUN TEST SCREENSHOT

Case-1

```
In [78]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
sentence_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this is normal string line
String is accepted
```

Case-2

```
In [80]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
sentence_validity_check.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this$is not @string
String is not accepted.
```

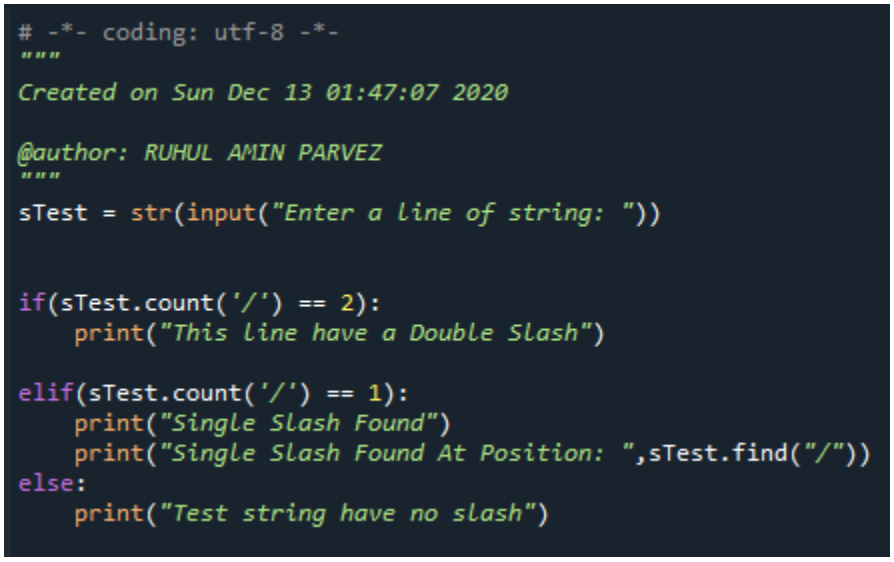
Lab Class – 5

Task-1 :- Program to search a single slash in a sentence then count and print the positions.

Code:

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 13 01:47:07 2020

@author: RUHUL AMIN PARVEZ
"""
sTest = str(input("Enter a line of string: "))


if(sTest.count('/') == 2):
   print("This line have a Double Slash")
elif(sTest.count('/') == 1):
   print("Single Slash Found")
   print("Single Slash Found At Position: ",sTest.find("/"))
else:
   print("Test string have no slash")
```

CODE SCREENSHOT



RUN TEST SCREENSHOT

Case-1

```
In [25]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
slash_count_with_position.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this is normal line
Test string have no slash
```

Case-2

```
In [27]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
slash_count_with_position.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this line have /single slash
Single Slash Found
Single Slash Found At Position:  15
```

Case-3

```
In [28]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
slash_count_with_position.py', wdir='C:/Users/RUHUL AMIN PARVEZ/
Downloads')

Enter a line of string: this line have //double slash
This line have a Double Slash
```

Lab Class – (6-7)

Task-1 :- Program to design a lexical analyzer.

Code:

```
#include <stdbool.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>
```

```
bool isDelimiter(char ch)
{
	if (ch == ' ' || ch == '+' || ch == '-' || ch == '*' ||
		ch == '/' || ch == ',' || ch == ';' || ch == '>' ||
		ch == '<' || ch == '=' || ch == '(' || ch == ')' ||
		ch == '[' || ch == ']' || ch == '{' || ch == '}')
		return (true);
	return (false);
}

bool isOperator(char ch)
{
	if (ch == '+' || ch == '-' || ch == '*' ||
		ch == '/' || ch == '>' || ch == '<' ||
		ch == '=')
		return (true);
	return (false);
}

bool validIdentifier(char* str)
{
	if (str[0] == '0' || str[0] == '1' || str[0] == '2' ||
		str[0] == '3' || str[0] == '4' || str[0] == '5' ||
		str[0] == '6' || str[0] == '7' || str[0] == '8' ||
		str[0] == '9' || isDelimiter(str[0]) == true)
		return (false);
	return (true);
```

```
}

bool isKeyword(char* str)
{
	if (!strcmp(str, "if") || !strcmp(str, "else") ||
		!strcmp(str, "while") || !strcmp(str, "do") ||
		!strcmp(str, "break") ||
		!strcmp(str, "continue") || !strcmp(str, "int")
		|| !strcmp(str, "double") || !strcmp(str, "float")
		|| !strcmp(str, "return") || !strcmp(str, "char")
		|| !strcmp(str, "case") || !strcmp(str, "char")
		|| !strcmp(str, "sizeof") || !strcmp(str, "long")
		|| !strcmp(str, "short") || !strcmp(str, "typedef")
		|| !strcmp(str, "switch") || !strcmp(str, "unsigned")
		|| !strcmp(str, "void") || !strcmp(str, "static")
		|| !strcmp(str, "struct") || !strcmp(str, "goto"))
		return (true);
	return (false);
}


bool isInteger(char* str)
{
	int i, len = strlen(str);

	if (len == 0)
		return (false);
```

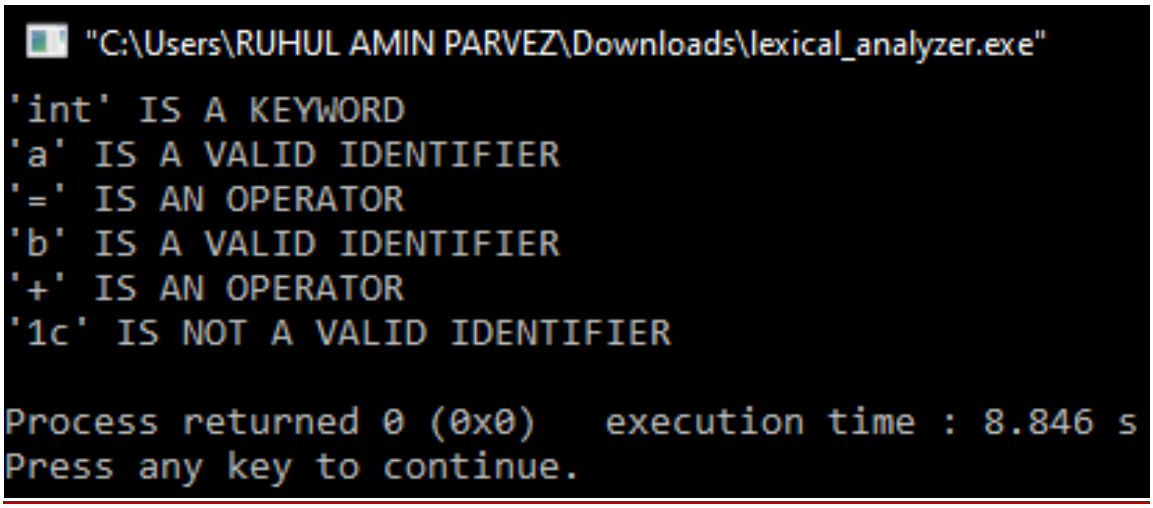
```
	for (i = 0; i < len; i++) {
		if (str[i] != '0' && str[i] != '1' && str[i] != '2'
			&& str[i] != '3' && str[i] != '4' && str[i] != '5'
			&& str[i] != '6' && str[i] != '7' && str[i] != '8'
			&& str[i] != '9' || (str[i] == '-' && i > 0))
			return (false);
	}
	return (true);
}

bool isRealNumber(char* str)
{
	int i, len = strlen(str);
	bool hasDecimal = false;

	if (len == 0)
		return (false);
	for (i = 0; i < len; i++) {
		if (str[i] != '0' && str[i] != '1' && str[i] != '2'
			&& str[i] != '3' && str[i] != '4' && str[i] != '5'
			&& str[i] != '6' && str[i] != '7' && str[i] != '8'
			&& str[i] != '9' && str[i] != '.' ||
			(str[i] == '-' && i > 0))
			return (false);
		if (str[i] == '.')
			hasDecimal = true;
	}
```

```
	return (hasDecimal);
}

char* subString(char* str, int left, int right)
{
	int i;
	char* subStr = (char*)malloc(
				sizeof(char) * (right - left + 2));

	for (i = left; i <= right; i++)
		subStr[i - left] = str[i];
	subStr[right - left + 1] = '\0';
	return (subStr);
}

void parse(char* str)
{
	int left = 0, right = 0;
	int len = strlen(str);

	while (right <= len && left <= right) {
		if (isDelimiter(str[right]) == false)
			right++;

		if (isDelimiter(str[right]) == true && left == right) {
			if (isOperator(str[right]) == true)
				printf("'%c' IS AN OPERATOR\n", str[right]);
```

```
                right++;
                left = right;
            } else if (isDelimiter(str[right]) == true && left != right
                    || (right == len && left != right)) {
                char* subStr = subString(str, left, right - 1);

                if (isKeyword(subStr) == true)
                    printf("'%s' IS A KEYWORD\n", subStr);

                else if (isInteger(subStr) == true)
                    printf("'%s' IS AN INTEGER\n", subStr);

                else if (isRealNumber(subStr) == true)
                    printf("'%s' IS A REAL NUMBER\n", subStr);

                else if (validIdentifier(subStr) == true
                        && isDelimiter(str[right - 1]) == false)
                    printf("'%s' IS A VALID IDENTIFIER\n", subStr);

                else if (validIdentifier(subStr) == false
                        && isDelimiter(str[right - 1]) == false)
                    printf("'%s' IS NOT A VALID IDENTIFIER\n", subStr);
                left = right;
            }
        }
        return;
}
```

```
int main()
{
	char str[100] = "int a = b + 1c; ";
	parse(str);
	return (0);
}
```

RUN TEST SCREENSHOT



Lab Class – 8

Task-1 :- Program to detect a perticular equation and change a character of that equation and print result.

Code:

```
"""
Created on Sun Dec 13 20:52:16 2020
@author: RUHUL AMIN PARVEZ
"""
```

equation = input("Enter a equation: ")

chng_one = input("Which letter do you want to change: ")

chng_two = input("Which letter do you want to put: ")

equation = equation.replace(chng_one, chng_two)

print(f"Now the equation is: ",equation)

CODE SCREENSHOT

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 13 20:52:16 2020

@author: RUHUL AMIN PARVEZ
"""


equation = input("Enter a equation: ")


chng_one = input("Which letter do you want to change: ")
chng_two = input("Which letter do you want to put: ")

equation = equation.replace(chng_one, chng_two)

print(f"Now the equation is: ",equation)
```

RUN TEST SCREENSHOT

Case-1

```
In [1]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/untitled0.py',
wdir='C:/Users/RUHUL AMIN PARVEZ/Downloads')

Enter a equation: x = y + 5

Which letter do you want to change: y

Which letter do you want to put: 6
Now the equation is:  x = 6 + 5
```

Case-2

```
In [30]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/
exchange_string.py', wdir='C:/Users/RUHUL AMIN PARVEZ/Downloads')

Enter a equation: x = y - 5

Which letter do you want to change: -

Which letter do you want to put: +
Now the equation is:  x = y + 5
```

Case-3

```
In [31]: runfile('C:/Users/RUHUL AMIN PARVEZ/Downloads/exchange_string.py',
wdir='C:/Users/RUHUL AMIN PARVEZ/Downloads')

Enter a equation: x = y + 5

Which letter do you want to change: x

Which letter do you want to put: z
Now the equation is:  z = y + 5
```