8th May 2023 (8/5/23)

Assignment No. 1

Question 1

→ Design a lexical analyser for a language. The lexical analyser should ignore redundant spaces, tabs and new lines. It should also ignore comments

→ Resource Type : C program

→ Program Logic:

check for new variable declarations to confirm identifiers
check for operators between identifiers and constants
check for special characters
check for comments
Display errors if any are not proper format

→ Program: C++

```
int main() {
    string code = "code here";
    string str = "";
    for(int x=0; x<code.length(); x++) {
        char c = code[x];
        if(checkOperator(c)) {
            cout << c << " is an operator\n";
            if(checkReal(str)) {
                cout << str << " is a constant";
                str = "";
                continue;
            }
```

```
if(checkKeyword(str)){
    cout << str << " is a keyword";
    str = "";
    continue;
}
if(!(str == ""))
    cout << str << "is a variable";
str = "";
continue;
```

→ Input:

```
float a = 60;
```

→ Output:

```
float is a keyword
a is a variable
= is an operator
; is a special char
60 is a constant
```

classmate
Date
Page

Question 2

→ Write a program to identify whether a given line is a comment or not

→ Resource Type : C++ program

→ Program Logic:

Read the input string
check whether the string starts with a '/' and check if the next character is '/' or '*'
If true, print true
Else it is not a comment

→ Program:

```
#include <iostream>
#include <string>
using namespace std;

int main () {
    string code = " ";
    for (int x = 0; x < code.length(); x++) {
        if (code[x-1] == '/')
            if (code[x] == '/' || code[x] == '*')
                cout << "comment";
    }
}
```

→ Input : // this is a comment
/* this is also */

→ Output :

Comment
Comment

Question 3

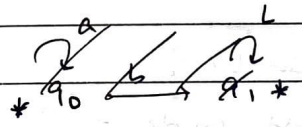
→ Write a program to recognise the strings
a*, a*L+, aLL

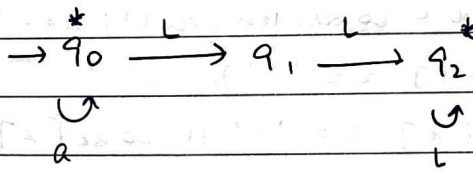
→ Resource Type : C++

→ Program Logic :

By using transition diagram, we verify the input
If the state is final, print true
Else print false

→ Program : State Diagram :






```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string input = " ";
    int state = 0;
```

```
for (int x=0; x<(input.length()); x++) { char i = input[x];
    if (state == 0) {
        if (i == 'a')
            state = 0;
        if (i == 'b')
            state = 1;
    }
    if (state == 1) {
        if (i == 'a')
            state = -1;
        if (i == 'b')
            state = 2;
    }
    if (state == 2) {
        if (i == 'a')
            state = -1;
    }
    if (state == 0 || state == 2)
        cout << "accepted";
}
```

→ Input : abbb

aabba

→ Output :

accepted

rejected

Question 4

→ Implement the lexical analyser using) Lex, Flex or other lexical analysing tools

→ Resource: Linux with putty

→ Program logic:

Read the input string
Check whether the string is identifier / key / symbol using the LEX Tool

→ Input: vi var.c

```
#include <stdio.h>
int main() {
    float a, b;
}
```

→ Output:

lex. lex.l
cc lex.yy.c
./a.out var.c
#include <stdio.h> is a PREPROCESSOR DIRECTIVE
Function
main (
)
BLOCK BEGINS
int is a KEYWORD
a IDENTIFIER
b IDENTIFIER
BLOCK ENDS