Bahria University



Lahore Campus

# Mid Term Exams (Spring-2024 Semester)

*Department of Computer Sciences*

|  |
| --- |
| Paper Show Date & Time:  Apr 24, 2024, 08:30AM – 11:30AM |

|  |  |  |  |
| --- | --- | --- | --- |
| **Exam Date & Time** | **Apr 26, 2024, 08:30AM – 11:30AM** | **Session** | **Spring 2024** |
| **Instructor Name** | **Mr. Muhammad Mudassar** | **Program/Semester:** | **BSCS/5A** |
| **Course Title** | **Compiler Construction** | **Course Code:** | **CSC 323** |
| **Time Allowed** | **90 mins** | **Max Marks** | **20** |

**NAME : AFFAN AHMAD ENROLLMENT: 03-134221-003**

**Instructions:**

**Read out the Instructions carefully.**

1. **Read the assignment carefully and attempt all the questions.**
2. **Understanding all questions is part of the exam.**
3. **Copied Assignment will straight away be awarded with *ZERO.***
4. **Timely submission is required, late submission will not be accepted at all.**
5. **Submission of the Midterm is to be done on LMS under Paper Section**
6. **For Output you need to take a full screen *Screenshot* and paste it in the given space.**
7. **The submission method is as follows; it should be strictly followed otherwise marks will be deducted.**

**Make a Folder to add this word solution file in the folder naming “*Name\_Enroll\_Midterm\_Section*” along with the .cpp files of each programming question. The .cpp file name should be saved according to the question no i.e “*Q1.cpp*” (no cpp file required for output question). Make .zip/.rar of the folder and upload it on the given link on LMS.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Student Name** ……………………………. **Enrollment Number**:………………………………

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation of CLO** | **Ques # / Part #** | **Marks** | **Obtained Marks** |
| **CLO3: Design and implement a compiler using a software engineering approach.** | 1 | 10 |  |
| 2 | 10 |  |
| **Total Marks** | | **20** |  |

**Question No 01: [1 + 1 + 2 + 2 + 4 = 10 marks]**

Write a C++ program that reads the text from (.txt) file consisting of more than 10,000 bytes of data. The program must be able to do the following:

* Load the data from .txt file using buffering concept with 3 fixed size arrays of 1024.
* Must be able to separate the words (tokens) and display the tokens without using any other array (except buffering arrays).
* Display the total number of tokens of the 2nd buffer (loaded only first time)
* Display the total no. of tokens in the whole text file.
* In last, store each word (token) as a new node at the end of link-list.

**Note**: Firstly, load data in 1st buffer, then 2nd and then in 3rd buffer. After 3rd buffer, control moves to 1st buffer again in case further text is there to read. The whole process continues until data ends in the text file.

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

const int BUFFER\_SIZE = 1024;

void processBuffer(const string& buffer,int count ) {

istringstream iss(buffer);

string token;

while (iss >> token) {

cout << token << endl;

count = count + 1;

}

}

int main() {

char buffer1[BUFFER\_SIZE];

char buffer2[BUFFER\_SIZE];

char buffer3[BUFFER\_SIZE];

bool useBuffer1 = true;

int count;

cout << "Enter text (press Ctrl+D on a new line to end input,if line not end try online compiler ):\n";

while (cin.good()) {

char\* currentBuffer = (useBuffer1) ? buffer1 : buffer2;

cin.getline(currentBuffer, BUFFER\_SIZE);

processBuffer(currentBuffer,count);

useBuffer1 = !useBuffer1;

}

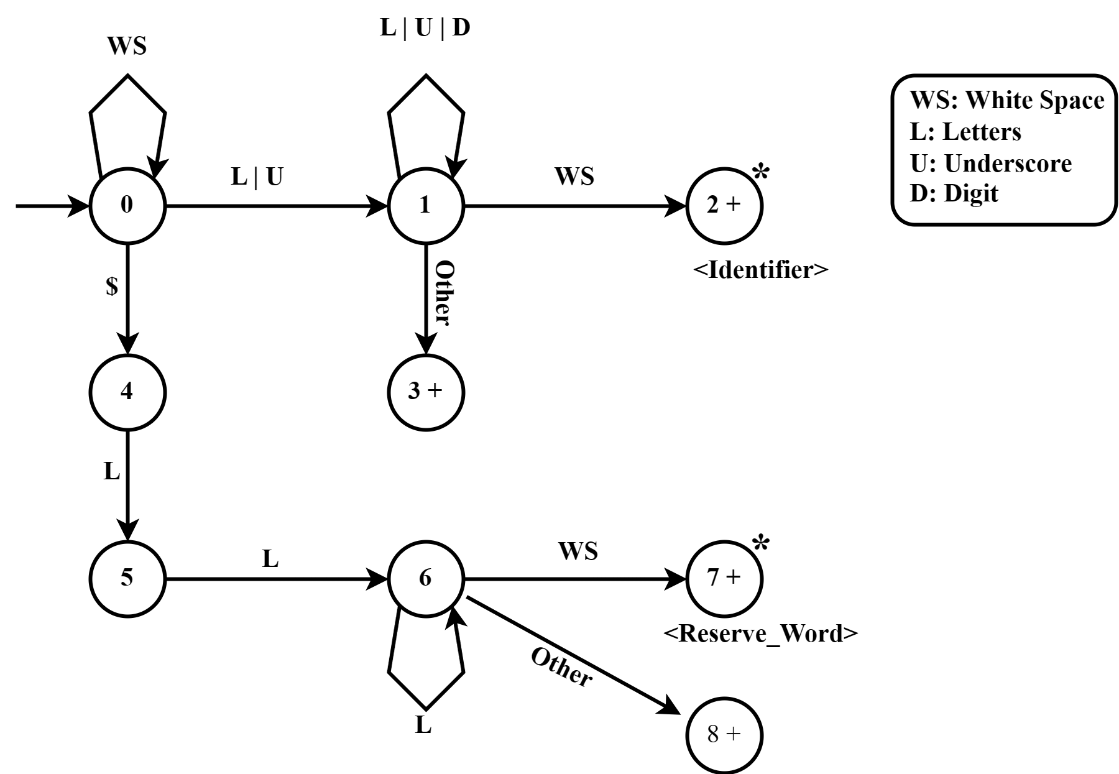
cout << "the total number of tokens are :" << count << endl;

return 0;

}

**Question No 02: [10 marks]**

Write a C++ program to construct lexical analyzer for the language illustrated by the given Deterministic Finite Automaton (DFA) using hard-coded method.



#include <iostream>

#include <sstream>

#include <string>

using namespace std;

int state = 0;

string l = "a";

string d = "3";

string array[] = { l, d, " ", "-" };

int i = 0;

void state0()

{

if (array[i] == " ")

{

state = 0;

i++;

}

else if (array[i] == "-" || array[i] == l)

{

state = 1;

i++;

}

else if (array[i] == "$")

{

state = 4;

}

}

int state1()

{

if (array[i] == l || array[i] == "-" || array[i] == d)

{

state = 2;

i++;

}

else if (array[i] == " ")

{

state = 2;

return 1;

}

else

{

state = 3;

return 1;

}

return 0;

}

void state4()

{

if (array[i] == l)

{

state = 5;

i++;

}

else

{

state = -1;

}

}

void state5()

{

if (array[i] == l)

{

state = 6;

}

else

{

state = -1;

}

}

int state6()

{

if (array[i] == l)

{

state = 6;

i++;

}

else if (array[i] == " ")

{

state = 7;

return 1;

}

else

{

state = 8;

return 1;

}

return 0;

}

int main()

{

for (int j = 0; j < 4; j++)

{

if (state == 0)

{

state0();

}

else if (state == 1)

{

cout << state1();

}

else if (state == 2 || state == 3)

{

cout << "string accepted" << endl;

break;

}

else if (state == 4)

{

state4();

}

else if (state == 5)

{

state5();

}

else if (state == 6)

{

cout << state6();

}

else if (state == 7 || state == 8)

{

cout << "string accepted" << endl;

break;

}

return 0;

}

}

**Do your own work, some One is watching.**

Best of Luck 👍