National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Compiler Construction	Course Code:	CS 4031
	Program:	BS (CS)	Semester:	Spring 2023
	Duration:	One hour	Total Marks:	35
	Paper Date:	11-Apr-2023	Weight:	
	Section:	6A and 8A	Page(s):	2
	Exam Type:	Mid-2		

Student : Name:_____ **Roll No.**_____

You can use an extra sheet for rough work. However do not attach the sheet with this paper

Question 1 (10+10 marks)

Give a translation scheme to convert a given C struct declaration into SQL create statement. For example consider the following declaration:

```
struct Student {
      int id;
      char grade;
}
```

This will be converted into the following statement:

```
create table Student (id:int, grade:char)
```

Use left-recursive CFG. Compute as attribute; do not print results. Use local variables; do not use global variables. Assume there are two data types only: int and char. Do not worry about arrays or pointers.

Solution

```
S -> struct id { L } [S.t="create table"+id.lex+"("+L.t+")"]
L -> L1 D          {L.t = L1.t + "," + D.t}
L -> D             {L.t = D.t}
D -> T id ;        {D.t = id.lex + ":" + T.t}
T -> int           {T.t = "int"}
T -> char          {T.t = "char"}
```

Question 2 (15 marks)

Give parsing code (C++) for the following CFG:

```
ROW -> R DL R'
DL -> DAT DL | ^
DAT -> D str D'
```

Following are the regular expressions for the tokens:

```
R -> <tr>              D -> <td>
R' -> </tr>            D' -> </td>
str -> char*
```

However you do not need to write code for the lexical analyzer

Solution

```
void ROW() {
      match(R);
      DL();
      match(R_);
}

void DL() {
      if (look == D) {
            DAT();
            DL();
      }
      else
            ;
}

void DAT() {
      match(D);
      match(str);
      match(D_);
}
```