1. **[20 marks]** The C programming language standard has two conventions that complicate the
design of a lexical analyzer for C.

1. **backslash continuation** If the backslash character \ occurs at the end of an input line, then
the next input line is taken as an immediate continuation of the first line. The backslash can
occur almost anywhere including in the middle of an identifier, reserved word or constant.

2. **trigraphs** ANSI C contains a brain-damaged mechanism for supporting keyboards that do
not contain all of the characters required to express C programs. A trigraph is a 3-character
sequence beginning with ?? that stands for some other single character.
The 9 ANSI trigraphs are shown below
   ```
   ??( is [ , ??) is ] , ??< is { , ??> is } , ??/ is \
   ??= is # , ??- is ~ , ??' is ^ , ??! is |
   ```
   These trigraphs may be used anywhere that the corresponding character could be used.

Discuss for these lexical features might be implemented in a lexical analyzer for C.

2. **[25 marks]** Describe the static semantic checks that a competent C compiler would perform
on the statements shown below

```c
... ...
10 if( ( ch = getchar() ) != EOF )
11 fprintf( stderr, "next char is %c", ch );
12 else
13 return ;
14 buffer[ bp++ ] = ch ;
15 for( K = 0 ; K < bp - 1 ; K++ )
16 if( ch == buffer[ K ]){
17 bp -- ;
18 break ;
19 }
```
3. [20 marks] For the declaration given below

```c
union bigU {
    unsigned char uchar ;
    struct {
        int ordinal ;
        double dNumb ;
    } stra ;
    struct {
        unsigned char cursor ;
        int xCoord, yCoord ;
        char * sptr ;
        double value ;
    } strb ;
    short clink[4] ;
};
```

Show how this structure would be laid out in memory using the space conserving Algorithm 2 that was discussed in lecture. Assume the size and alignment constraints in the table below.

<table>
<thead>
<tr>
<th>type</th>
<th>size</th>
<th>align</th>
<th>type</th>
<th>size</th>
<th>align</th>
</tr>
</thead>
<tbody>
<tr>
<td>char *</td>
<td>32</td>
<td>32</td>
<td>unsigned char</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>64</td>
<td>short</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>int</td>
<td>32</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. [15 marks] Show the symbol and type table entries that a typical compiler would create for the declarations in Question 4.

5. [20 marks] Which of the following four grammars are LL(1)? Justify your answers. Answers without justification will get partial credit. ( λ is the empty string ).

1. $$S \rightarrow A \ A 'b' \ 'c'$$
   $$A \rightarrow 'a'$$
   $$A \rightarrow \lambda$$
   $$A \rightarrow \lambda$$
   $$B \rightarrow 'b'$$
   $$B \rightarrow \lambda$$

2. $$S \rightarrow A \ B B A$$
   $$A \rightarrow 'a'$$
   $$A \rightarrow \lambda$$
   $$A \rightarrow \lambda$$
   $$B \rightarrow 'b'$$
   $$B \rightarrow \lambda$$

3. $$S \rightarrow A 'b'$$
   $$A \rightarrow 'a'$$
   $$A \rightarrow B$$
   $$A \rightarrow \lambda$$
   $$B \rightarrow 'b'$$
   $$B \rightarrow \lambda$$

4. $$S \rightarrow 'a' S 'e'$$
   $$S \rightarrow B$$
   $$B \rightarrow 'b' B 'e'$$
   $$B \rightarrow C$$
   $$C \rightarrow 'c' C 'e'$$
   $$C \rightarrow 'd'$$