1. **Problem 1**

Which compiler phase (e.g. scanner, parser, constrainer, code generator) would be responsible for catching each of the following errors, respectively? Explain why.

1. An illegal character.
2. Premature end-of-line.
3. A undeclared variable.
5. Division by zero.

2. **Problem 2**

(Chapter 1, Exercise 1.5)

1. In your local implementation of C, what is the limit on the size of integers? What happens in the event of arithmetic overflow? What are the implications of size limits on the portability of programs from one machine/compiler to another?

2. How do the answers to these questions differ for Java? For Ada? For Pascal? (You may need to find a manual.)
3. Problem 3

Consider the following simplified version of English Grammar:
<sentence> ::= <noun phrase> <verb phrase>.
<noun phrase> ::= <determiner> <noun> | <determiner> <noun> <prepositional phrase>
<verb phrase> ::= <verb> | <verb> <noun phrase> | <verb> <noun phrase> <prepositional phrase>
<prepositional phrase> ::= <preposition> <noun phrase>
<noun> ::= boy | girl | dog | ball | photo | feather
<determiner> ::= a | the
<verb> ::= saw | touched | kicked | took
<preposition> ::= by | with

a). Draw a Derivation Tree of the following input:
the boy touched the dog with a feather.

b). Is this an ambiguous grammar? why?

4. Problem 4

Given the following LL(1) grammar:
1. P -> S $ $
2. S -> ( S ) S
3. S -> [ S ] S
4. S ->

and the following input:
( ( ) [ ] ) [ ] $ $

1. Draw the Parse Table for the grammar.

2. Draw the LL(1) Parsing Traces of the input sentence above, with the format similar to that in lecture slides (04page#13).

3. Which has higher precedence? "[" or "("?