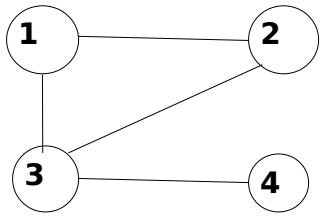
	Course Name:	Compiler Construction	Course Code:	CS-402
	Program:	BS (CS)	Semester:	Fall 2018
	Duration:	60 Minutes	Total Marks:	25
	Paper Date:	16-Nov-2018	Weight	
	Section:	ALL	Page(s):	2
	Exam Type:	Midterm-II		

Student : Name:_____ Roll No._____
Section:_____

Instruction/Notes: Solve Q1 on page 1, and Q2 on page 2 and 3. Only the first three pages will be marked!

Question 1 (10 marks)

An undirected graph can be represented in many ways. One common method is to use an adjacency matrix; another is to use an adjacency list. Moreover we can use XML-like language for this purpose. Consider the following graph for example:



We can represent this graph as follows:

```
<graph>
      <node> 1, 2, 3, 4 </node>
      <edge>
            (1,2), (1,3), (2,3), (3,4)
      </edge>
</graph>
```

The same graph can be represented using an adjacency matrix as follows:

	1	2	3	4
1		1	1	
2	1		1	
3	1	1		1
4			1	

Give a translation scheme to convert the XML-like representation of any given graph into an equivalent adjacency matrix representation. Use the following regular definition and CFG:

```
// Regular definition:

GS -> < g r a p h >
GE -> < / g r a p h >
NS -> < n o d e >
NE -> < / n o d e >
ES -> < e d g e >
EE -> < / e d g e >
num -> digit+
digit -> 0 | 1 | ... | 9


// CFG:

G -> GS N E GE
N -> NS NL NE
NL -> NL , num | num
E -> ES EL EE
EL -> EL , P | P
P -> ( num , num )
```

You may use global variables. Do not change the regular definition or the CFG!

Question 2 (5+10 marks)

a) Draw a parse tree for the following translation scheme. Use the string "123" as input. Show the attribute values.

```
N -> N1 digit {N.v = N1.v * 10 + int(digit.lex)}
N -> digit {N.v = int(digit.lex)}
```

b) Eliminate left recursion from the above translation scheme.