# COSC202 Midterm Test -- Programming Languages I (Full Marks 40)

1. Recursive descent parsing. (12 points)
1.1 Express the following context-free grammar by a syntax chart.

$S \rightarrow DSa \mid a$
$D \rightarrow cD \mid d$

1.2 Attach a selection set at each forking point.

1.3 Trace the syntax chart in a recursive descent manner for the input string ccdaa!, where "!" is an end-marker. Follow the example in page 17 for the style of the trace.

2. The statement "if B1 then if B2 then S1 else S2" can be interpreted as

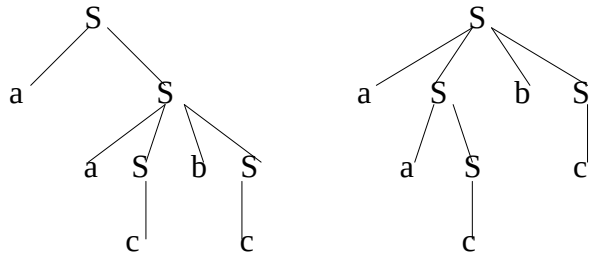
(1) if B1 then (if B2 then S1 else S2)

(2) if B1 then (if B2 then S1) else S2.

If we name "if B then" by "a" and "else" by "b", and "S" by "c", then the problem can be modelled by the context-free grammar

1. $S \rightarrow aS$, 2. $S \rightarrow aSbS$, 3. $S \rightarrow c$.

The above ambiguous statement can be parsed by the following two trees.




2.1 Draw all possible derivation trees for the string a a a a c b c b c (4 points)

2.2 How many derivation trees are there for the string $a^{2n}(cb)^{n}c$? (4 points)

2.2 The following program produces different results depending on the above two interpretations. (4 points)

x:=1; y:=0; if x >0 then if y > 0 then x:=x+1 else y:=y+1;

Show the values of x and y at the end of the program for each interpretation. If we change the initialisation part to x:=0; y:=0;, are the results the same? Give the reason. Is there any initialisation that brings the same results under the two interpretations?

3. The object code for the following PL/0 program is given below.

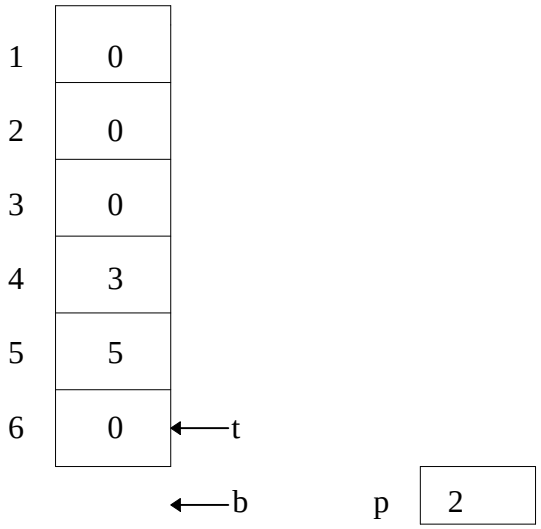
```
CONST M = 3, N = 5;
VAR X, Y, Z;
PROCEDURE MULTIPLY;
  VAR A, B;
BEGIN
   A:=X; B:=Y; Z:=0;
   WHILE A > 0 DO BEGIN
      Z:=Z+B;
      A:=A-1;
    END;
END;
BEGIN
    X:=M; Y:=N;
    CALL MULTIPLY;
END.
```

```
0   JMP   0   23
1   JMP   0   2
2   INT   0   5
3   LOD   1   3
4   STO   0   3
5   LOD   1   4
6   STO   0   4
7   LIT   0   0
8   STO   1   5
9   LOD   0   3
10  LIT   0   0
11  OPR   0   12
12  JPC   0   22
13  LOD   1   5
14  LOD   0   4
15  OPR   0   2
16  STO   1   5
17  LOD   0   3
18  LIT   0   1
19  OPR   0   3
20  STO   0   3
21  JMP   0   9
22  OPR   0   0
23  INT   0   6
24  LIT   0   3
25  STO   0   3
26  LIT   0   5
```

```
27   STO    0     4
28   CAL    0     2
29   OPR    0     0
```

3.1 Give a comment on the right of each machine instruction of the object code, following the example in page 36. Give a comment for each JPC instruction also. (8 points)

3.2 The following is a snapshot of the stack s after the “CAL” instruction is executed. (8 points)




Following this example, give a snapshot of the stack and registers after the “OPR” at line 15 is executed for the third time. Note that undefined array elements are initialised to 0.