SE 4367 Homework #4, ASTs and CFGs

1. Draw the abstract syntax trees for the following predicates, AST(p_r), where a, b, c, and d are Boolean variables:

a) a + cd
b) abc
c) a + !(bc) + d
d) a!bc + ab!d
e) a + !b(!a!c + d)

Notation

For AND, you could use *, ∧, AND, …

For OR, you could use +, ∨, OR, …

For NOT, you could use !, ~, NOT, …

Note the precedence of Boolean operators

- **parentheses**
- **NOT**
- **AND**
- **OR**

Mathematically speaking, left-to-right isn't necessary, but it is the convention…

2. Program P1 CFG.

a) Identify the basic blocks for the following program P1 written in pseudo-code.

b) Draw the control flow graph.

```
 Program P1
1) integer A, B;
2) input (A);
3) if (A == 0)
4) {
5)       B = A + 1;
6) }
7) else
8) {
9)       B = A - 2;
10)}
11)output (A,B);
12)end;
```

3. Program P2 CFG.

a) Identify the basic blocks for the following program P2 written in pseudo-code.

b) Draw the control flow graph.

Program P2

```
1) integer A, B;
2) input (A);
3) B = 1;
4) while (int i=1; i<=A; i++)
5) {
6)       B = B * i;
7) }
8) output (A,B);
9) end;
```

4. Program P3 CFG.
a) Identify the basic blocks for the following program P3 written in pseudo-code.
b) Draw the control flow graph.

Program P3

```
1)   integer A, B;
2)   input (A);
3)   if (A > 7)
4)         B = 1;
5)   else
6)   {
7)         B = 2;
8)         if (A < 2)
9)               B = 3;
10)   } // end else A>7
11)  while (int i=1; i<=A; i++)
12)  {
13)        if (B<0)
14)              B = B + 4;
15)        else
16)              B = B - 5;
17)  } // end for loop
18)  output (A,B);
19)  end;
```

5. Program P4 CFG.

a) Identify the basic blocks for the following program P4 written in pseudo-code. Note the post-test loop!

b) Draw the control flow graph.

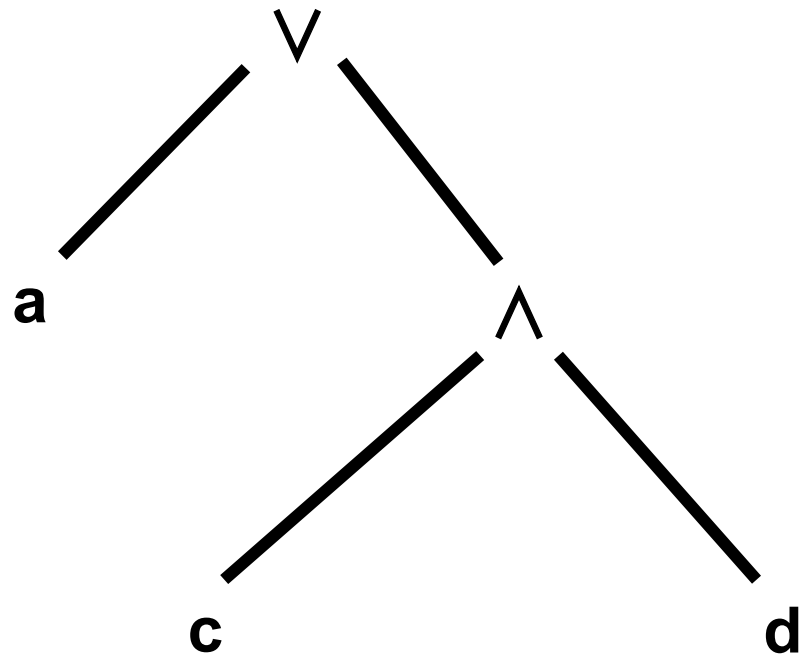
Program P4

```
1) integer A, B;
2) input (A);
3) B = 1;
4) do {
5)       B = B * A;
6)       A = A - 1;
7) } while (A<=0);
8) output (A,B);
9) end;
```

$$4.1a)\ a + cd$$

Precedence

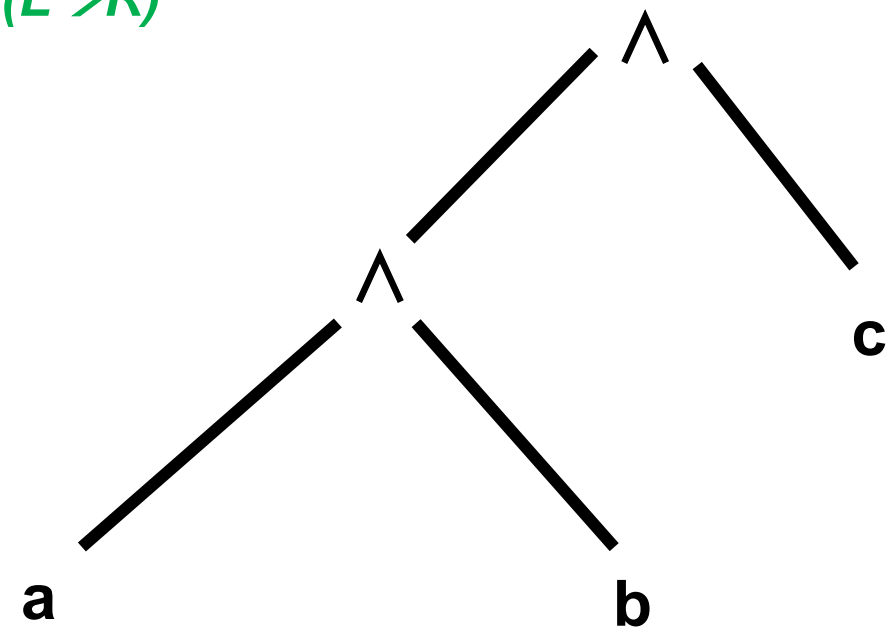
1) ***AND***
2) ***OR***



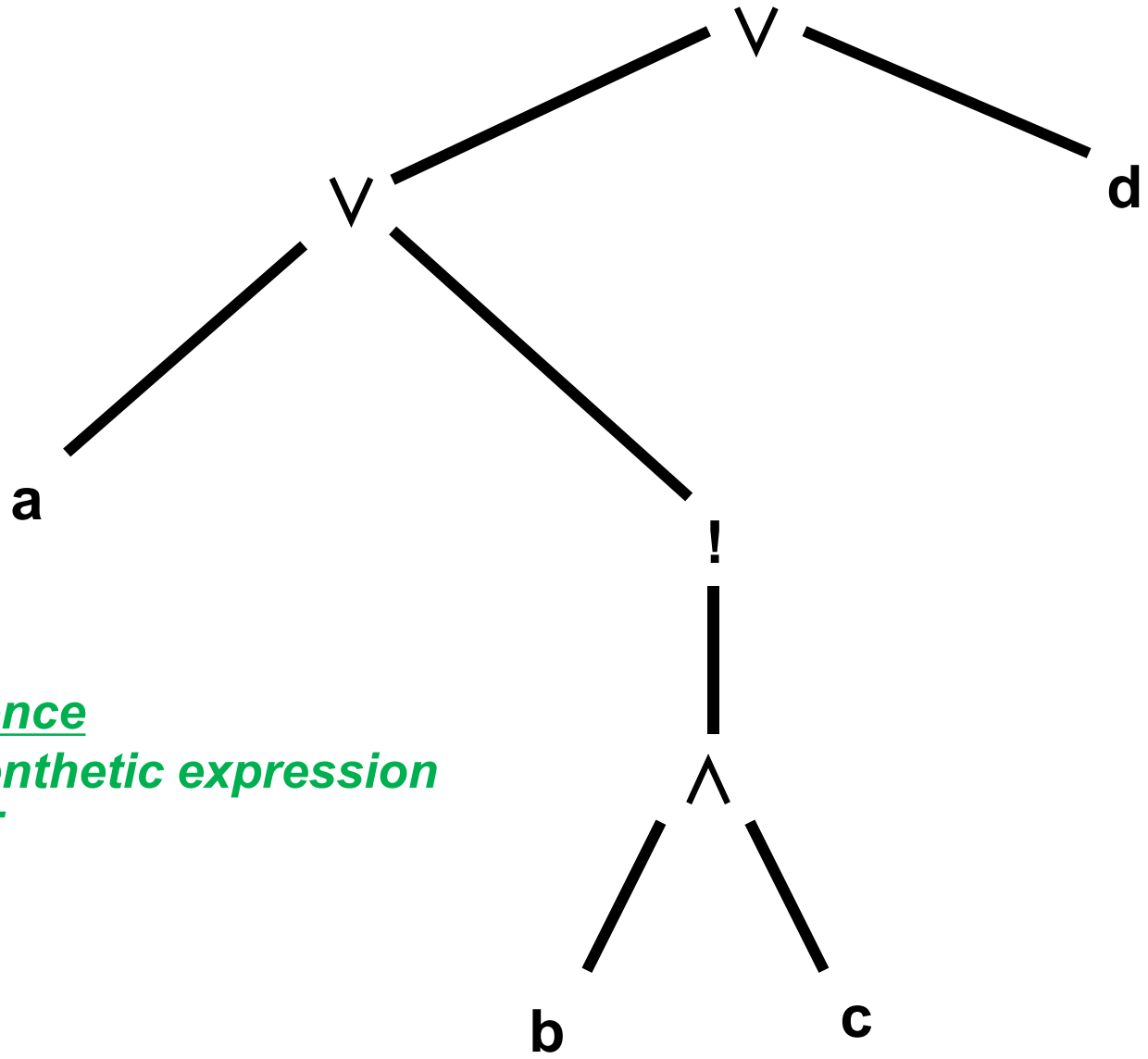
4.1b) abc

Precedence

1) AND (L→R)



$$4.1c)\ a + !(bc) + d$$



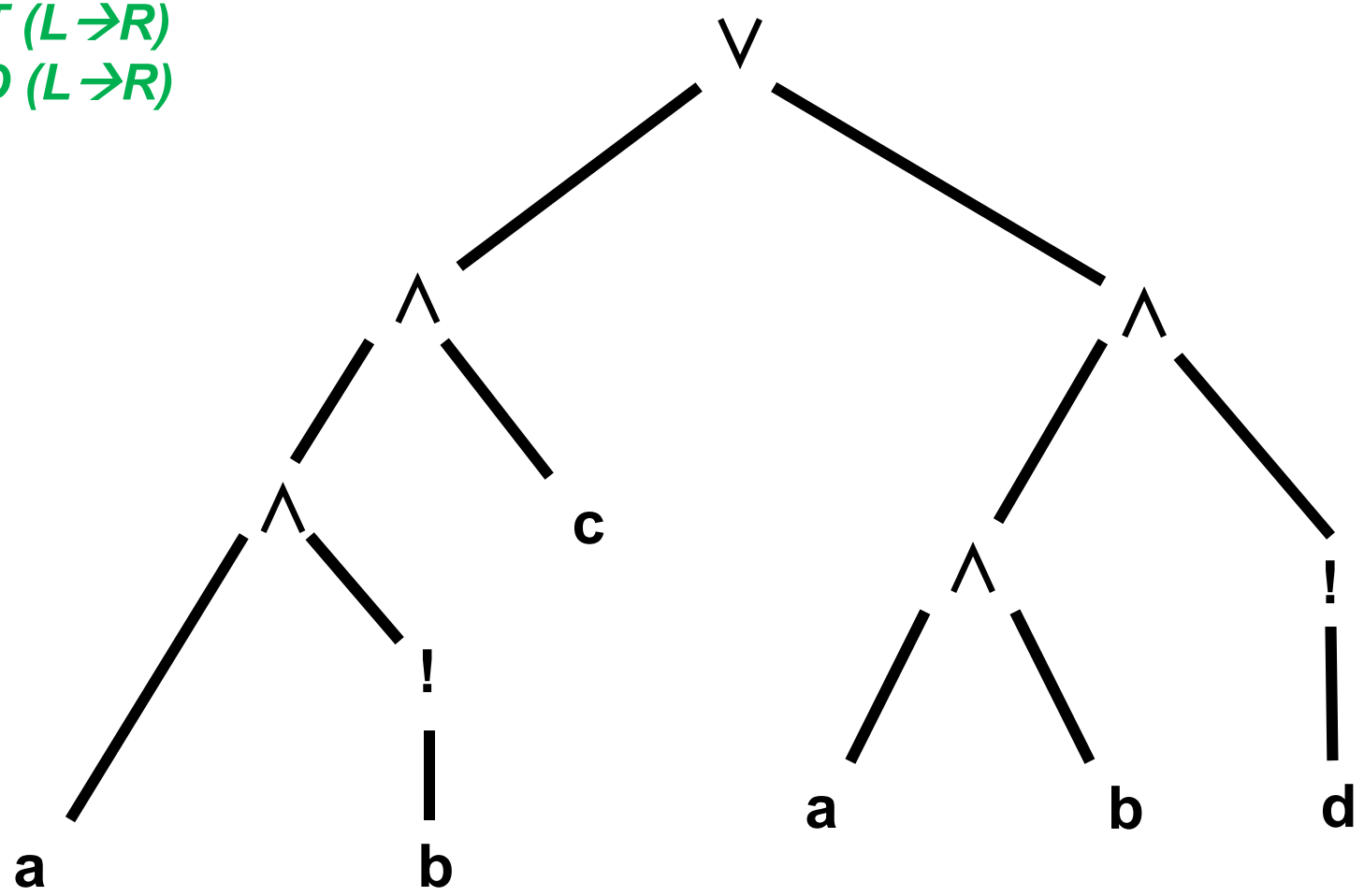
Precedence

1) *parenthetic expression*
2) *NOT*
3) *OR*

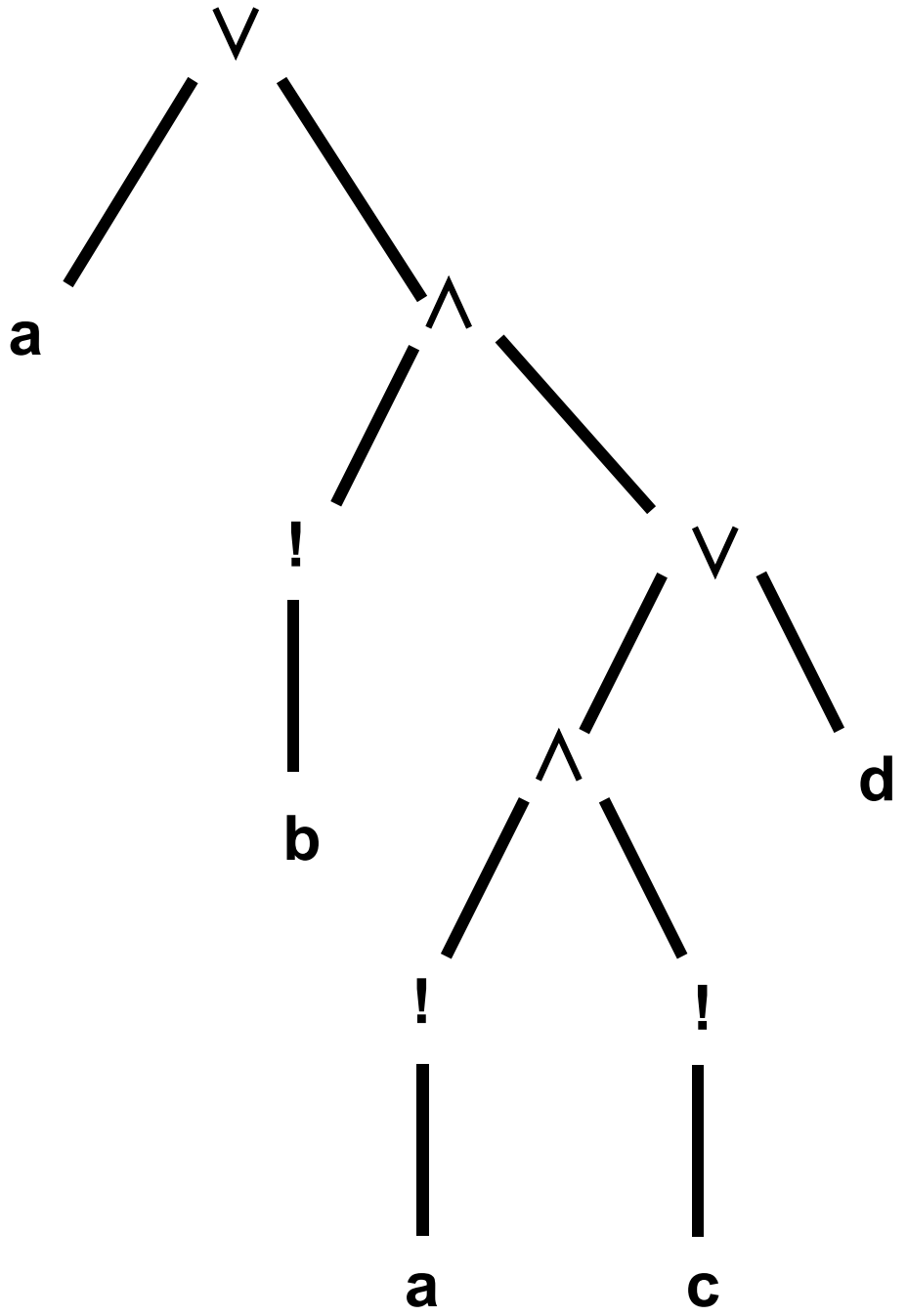
4.1d) a!bc + ab!d

Precedence

1) ***NOT (L→R)***
2) ***AND (L→R)***
3) ***OR***



$4.1e)\ a + !b(!a!c + d)$



Precedence

1) *parenthetic expression*
2) *NOT*
3) *AND (L→R)*
4) *OR (L→R)*

2. Program P1 CFG.

a) Identify the basic blocks for the following program P1 written in pseudo-code.

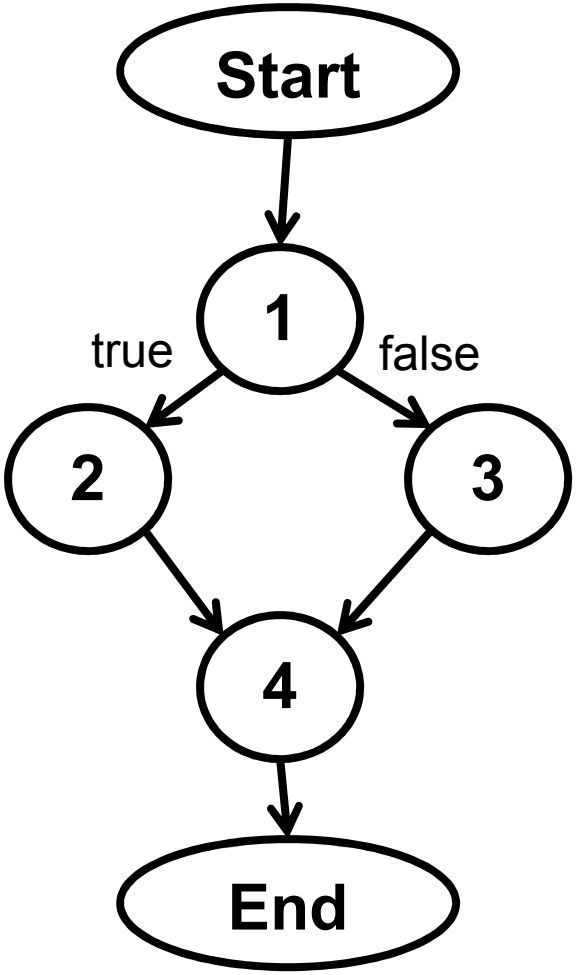
b) Draw the control flow graph.

```
Program P1
1) integer A, B;
2) input (A);
3) if (A == 0)
4) {
5)       B = A + 1;
6) }
7) else
8) {
9)       B = A - 2;
10)}
11)output (A,B);
12)end;
```

```
 Program P1
1) integer A, B;
2) input (A);
3) if (A == 0)
4) {
5)       B = A + 1;
6) }
7) else
8) {
9)       B = A - 2;
10)}
11)output (A,B);
12)end;
```

Basic blocks

1 – 1, 2, 3 (4)
2 – 5 (6, 7, 8)
3 – 9 (10)
4 – 11 (12)



3. Program P2 CFG.

a) Identify the basic blocks for the following program P2 written in pseudo-code.

b) Draw the control flow graph.

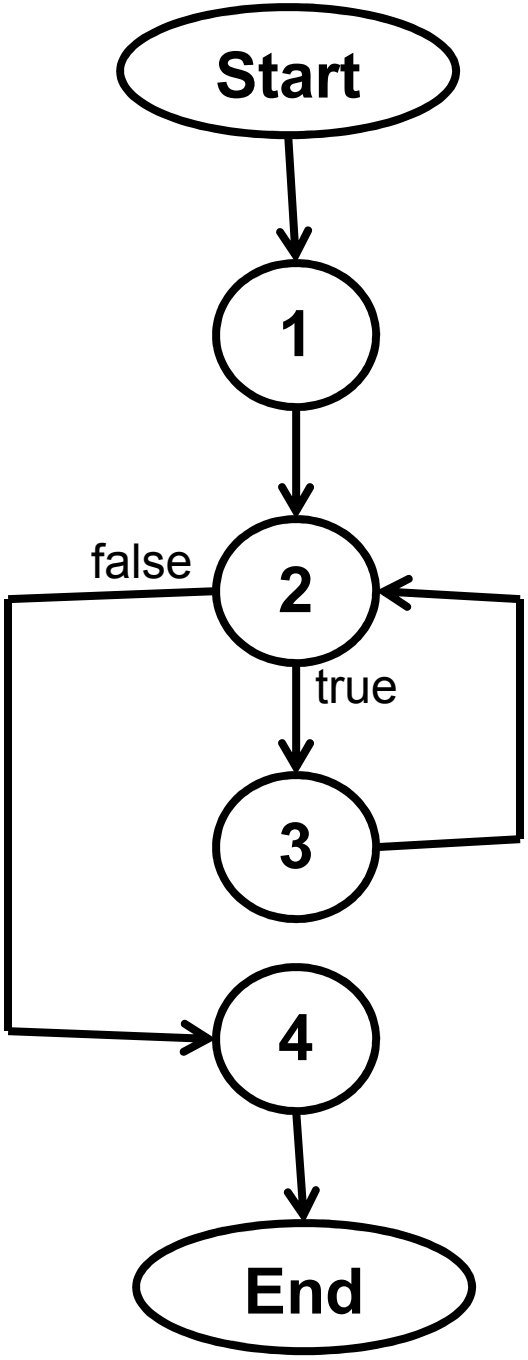
Program P2

```
1) integer A, B;
2) input (A);
3) B = 1;
4) while (int i=1; i<=A; i++)
5) {
6)       B = B * i;
7) }
8) output (A,B);
9) end;
```

```
 Program P2
1) integer A, B;
2) input (A);
3) B = 1;
4) while (int i=1;
   i<=A; i++)
5) {
6)       B = B * i;
7) }
8) output (A,B);
9) end;
```

Basic blocks

1 – 1, 2, 3
2 – 4 (5)
3 – 6 (7)
4 – 8 (9)



4. Program P3 CFG.
a) Identify the basic blocks for the following program P3 written in pseudo-code.
b) Draw the control flow graph.

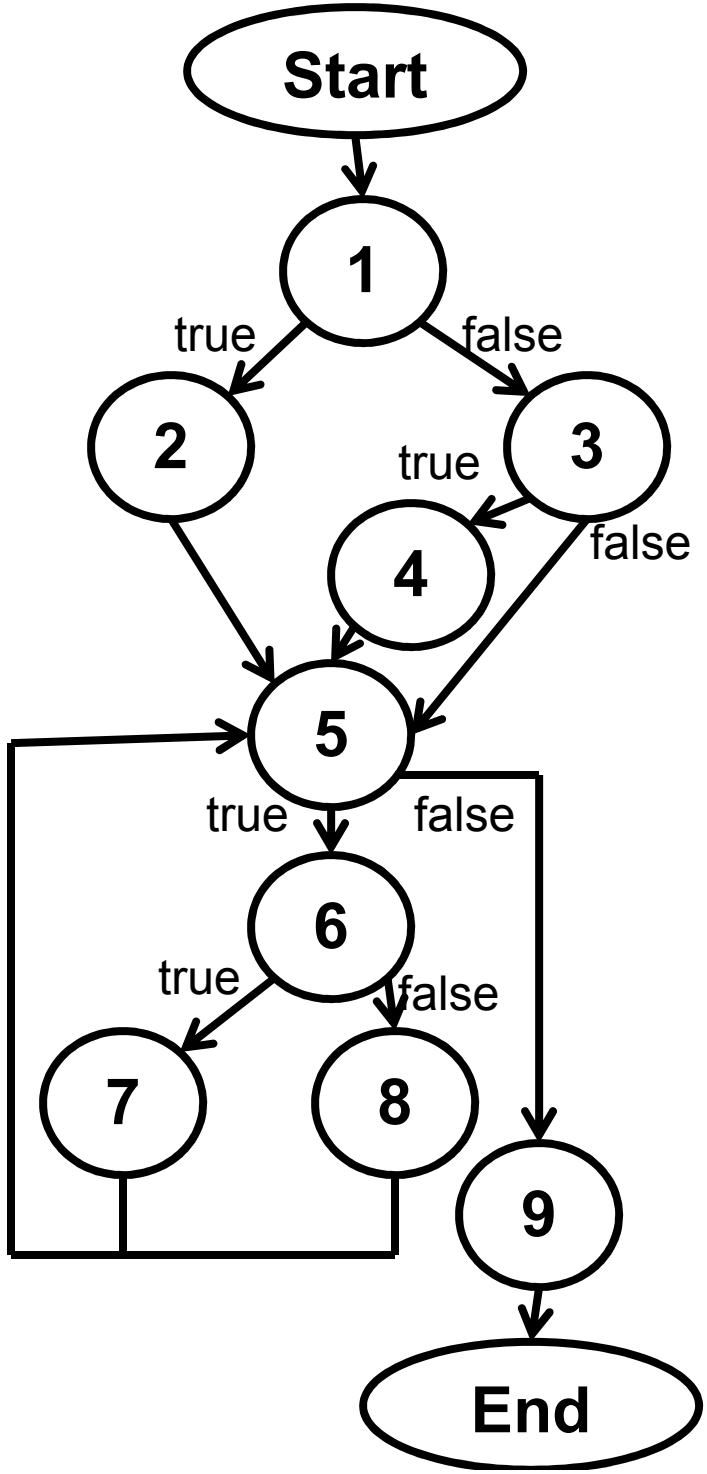
Program P3

```
1)   integer A, B;
2)   input (A);
3)   if (A > 7)
4)         B = 1;
5)   else
6)   {
7)         B = 2;
8)         if (A < 2)
9)               B = 3;
10)   } // end else A>7
11)  while (int i=1; i<=A; i++)
12)  {
13)        if (B<0)
14)              B = B + 4;
15)        else
16)              B = B - 5;
17)  } // end for loop
18)  output (A,B);
19)  end;
```

```
 Program P3
1)    integer A, B;
2)    input (A);
3)    if (A > 7)
4)          B = 1;
5)    else
6)    {
7)          B = 2;
8)          if (A < 2)
9)                B = 3;
10)   } // end else A>7
11)   while (int i=1; i<=A;
      i++)
12)   {
13)         if (B<0)
14)               B = B + 4;
15)         else
16)               B = B - 5;
17)   } // end for loop
18)   output (A,B);
19)   end;
```

Basic blocks

1 – 1, 2, 3
2 – 4 (5, 6)
3 – 7, 8
4 – 9 (10)
5 – 11 (12)
6 – 13
7 – 14 (15)
8 – 16 (17)
9 – 18 (19)



5. Program P4 CFG.

a) Identify the basic blocks for the following program P4 written in pseudo-code. Note the post-test loop!

b) Draw the control flow graph.

Program P4

```
1) integer A, B;
2) input (A);
3) B = 1;
4) do {
5)       B = B * A;
6)       A = A - 1;
7) } while (A<=0);
8) output (A,B);
9) end;
```

Program P4

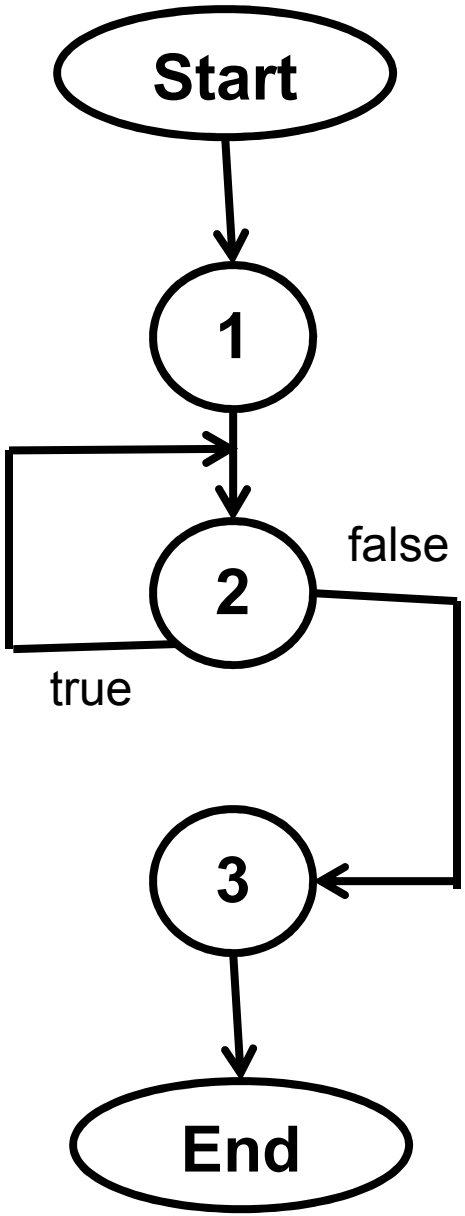
```
1)  integer A, B;
2)  input (A);
3)  B = 1;
4)  do {
5)      B = B * A;
6)      A = A – 1;
7)  } while (A<=0);
8)  output (A,B);
9)  end;
```

Basic blocks

1 – 1, 2, 3
2 – (4) 5, 6, 7
3 – 8 (9)

Two entries into do begins a block

Two exits out of do-while ends the block (like an if)



Grading Rubric

Each of the five problems is worth 20 points.

For #1, each AST is worth 4 points.

For #2-5, each of the eight (4*a,b) parts above is worth 10 points each.

Each basic block and each node & edge is worth a proportional part of its 10 points. For example,

- **if there are 5 basic blocks for part a, correctly defining each is worth 2 points apiece**
- **if there are 5 nodes and 5 edges in the CFG for part b, each node and edge correctly drawn is worth 1 point**

Formatting Submissions

In the file name, include:

- **class**
- **assignment identifier**
- **your name (or team's name)**
  - **e.g., se4367a01jdoe**

In the file (or hardcopy) submitted, include the class, assignment, and name information at the top.

Minus 5 points per violation. Potentially 30 points off for formatting mistakes!