Due date: Due on **Apr 19th (Monday) 10am in class.**
Use the following procedure for both problems:

```java
1: public void printDate (int day, boolean leap_year) {
2:    int month = 1;
3:    int days_per_month = 31;
4:    while (day > days_per_month) {
5:       day = day - days_per_month;
6:       month++;
7:       if (month==3 || month==5 || month==7 || month==8 || month==10 || month==12)
8:          days_per_month = 31;
9:       else if (month==4 || month==6 || month==9 || month==11)
10:          days_per_month = 30;
11:       if (month==2)
12:          if (leap_year)
13:             days_per_month = 29;
14:          else
15:             days_per_month = 28;
16:    }
17:    System.out.println(day + "-" + month);
18: }
```

1. Draw the control flow graph for printDate. Use the conventions we covered in class; a composite condition needs to be split into predicates.

2. Cyclomatic complexity
   
   (a) Compute the cyclomatic complexity of printDate using the formula: $CV = e - n + 2$
   
   (b) List the predicate nodes
   
   (c) Compute the cyclomatic complexity of printDate using the formula: $CV = \#\text{predicate nodes} + 1$. 

1