

```

/**
 * Format has 3 format methods, one each for
 * int, double and String. The format methods
 * return a String representing the original input
 * fitted into a column of specified width and justified
 * either LEFT or RIGHT. In addition, the double is rounded
 * to a specified number of decimal places. For example, the
 * following code ...
 * <pre>
 *     String hb = "Happy Birthday" ;
 *     for(int i = 1 ; i < 11 ; i++) {
 *         double d = 100.0/i ;
 *         String s = Format.format(i, 5, LEFT) +
 *                 Format.format(hb.substring(0, i), 12, LEFT) +
 *                 Format.format(d, 2, 7, RIGHT) ;
 *         System.out.println(s) ;
 *     }
 * </pre>
 * would produce the following little table ..
 * <pre>
 *1   H           100.00
 *2   Ha          50.00
 *3   Hap         33.33
 *4   Happ        25.00
 *5   Happy       20.00
 *6   Happy       16.67
 *7   Happy B    14.29
 *8   Happy Bi   12.50
 *9   Happy Bir  11.11
 *10  Happy Birt 10.00
 *
 * </pre>
 */
public class Format {
    /**
     * format resulting string with LEFT justification
     */
    public static final int LEFT = -1 ; // push to left
    /**
     * format resulting string with RIGHT justification
     */
    public static final int RIGHT = 1 ; // push to right

    /**
     * format(int n,int w,int justify) <BR>
     * format int n in a column w wide and justify LEFT or RIGHT
     */
    public static String format(int n, int w, int justify) {
        String s = String.valueOf(n) ;
        while (s.length() < w) {
            if (justify == LEFT) s = s + " " ;
            else s = " " + s ;
        }
        return s ;
    }

    /**
     * format(String s, int w, int justify) <BR>
     * format String s in a column w wide and justify LEFT or RIGHT
     * trim s to length w by cutoff at right if necessary
     */
    public static String format(String s, int w, int justify) {
        String ss = s ;
        while (ss.length() < w) {
            if (justify == LEFT) ss = ss + " " ;
            else ss = " " + ss ;
        }
        return ss.substring(0, w) ;
    }
}

/**

```

```
* format(double d, int aft, int w, int justify) <BR>
* format double d using aft decimal places in a column w wide and
* justify LEFT or RIGHT
*/
public static String format(double d, int aft, int w, int justify) {
    // convert rounded d to String first
    String s = String.valueOf(d + 5.0*Math.pow(10, -(aft+1))) ;
    int p = 0 ;
    boolean found = false ;
    while (!found) { // find where '.' is
        if (s.charAt(p) != '.') p++ ;
        else found = true ;
    }
    String fore = s.substring(0, p) ;
    String after = s.substring(p+1, s.length()) ;
    while (after.length() < aft) after = after + "0" ; // pad 0's if necessary
    after = after.substring(0, aft) ;
    String result = fore + "." + after ;
    while (result.length() < w) {
        if (justify == LEFT) result = result + " " ;
        else result = " " + result ;
    }
    return result ;
}

public static void main(String args[]) {
    String hb = "Happy Birthday" ;
    for(int i = 1 ; i < 11 ; i++) {

        double d = 100.0/i ;
        String s = Format.format(i, 5, LEFT) +
            Format.format(hb.substring(0, i), 12, LEFT) +
            Format.format(d, 2, 7, RIGHT) ;
        System.out.println(s) ;
    }
}
}
```