

COP 3223 Recitation: Functions

Problem A: Not Perfectly Random Jukebox

You and your roommate just got an old mini juke box from a friend. You want to program it so that it plays random songs, but never plays the same song twice in a row. Luckily, your roommate, who is also in COP 3223, has written a main function for this program and just needs a tiny bit of help from you. Help your roommate by writing a function that takes in two integer parameters - *n*, the number of songs in the jukebox and *last*, the number of the last song played by the jukebox. Your function should generate random integers in between 0 and *n*-1 and return the first one of these not equal to *last*. The function prototype is given to you below and you can download the program to add your function to:

```
// Pre-condition: n > 1, 0 <= last < n
// Post-condition: Returns a random integer in between 0 and
//                  n-1, not equal to last.
int nextSong(int n, int last);
```

Problem B: Jukebox Battery

Your jukebox has a visual ascii display. You come up with the clever idea of having it display a bar representing how much battery is left. Using the interface to the display, it turns out that you just have to write a void function in C to print out what to display. Your function will take in a percentage and you'll display a bar corresponding to that percentage, with labels on the left-hand side. The function prototype is below. Write your own main to test the function.

```
// Pre-condition: 0 <= perc <= 100, c is a printable character
// Post-condition: Prints a bar corresponding to perc using the
//                  character c with width 7 and percentage labels
void printBatteryStatus(int perc, char c);
```

Round *perc* to the nearest 5% (22% goes to 20% and 43% goes to 45%, for example), and then draw a bar of the character *c*. Here is what should get printed for *perc* = 22% and *c* = '*':

```
100
 95
 90
 85
 80
 75
 70
 65
 60
 55
 50
 45
 40
 35
 30
 25
 20  *
 15  *
 10  *
  5  *
    Battery
```