SCC110: Software Development Term 1. C Programming.

Problem Set 6: Fruit Machine with Functions

The Goal
This week's lab aims to:

- Extend your familiarity with declaring and using functions
- Think about refactoring code to use functions (the fruit machine solution you started last week)
- Practice passing arrays into functions

1.1 SCC.110 Fruit Machine

The SCC.110 fruit machine has 3 reels, each containing the following symbols:

1. a bell
2. an orange
3. a cherry
4. a horseshoe

Your goal is to successfully model the operation of the fruit machine, specifically:

1. Spin the reels (i.e. pick which symbol is to be shown on each reel)
2. Print out the reels (we suggest as text, e.g. `bell - orange - cherry`)
3. Detect whether the player has won (the player wins when two or more symbols match, jackpot is when all three symbols match)

You should think carefully about how to model your solution as a set of functions.

1.2 Function decomposition

Refactor your fruit machine implementation to use functions. For example, you could use a function to generate a new value when you 'spin a reel'; and a further function to display the state of the reel (what symbol is showing), and so forth.

- Your solution must have at least 3 functions including `main`().
- Make good use of the return value from functions and parameters to pass values into and out of your functions as necessary.
- You are **not** to use *global variables*.

1.3 Optional/desirable: Variable number of reels using arrays

If you have your fruit machine working and you have enough time left in this week's lab, then start this challenge—it'll really help you get to grips with passing arrays into functions.

Extend your previous solution to use an array to represent your 3 reels. Refactor your program's functions so they work with the array parameter where necessary. Your game should ideally be extended to allow for a variable number of reels (from 1 to 9).

- Your solution should have a function that takes an array of reels as a parameter (e.g. randomising the reels, printing out the reels each round of the game)
- Your solution should allow for a variable number of reels

- Ideally, your program will allow for the user to enter the number of reels, and this should not compromise the 'jackpot' and winnings calculation function

Hacker edition

If you're up to date with all the tasks, and the hacker editions, then undertake this advanced task- this'll also give you knowledge you'll find useful if you choose a C project in Summer Term.

The advanced task this week extends last week's task by refactoring your functions to use pointers where appropriate to pass parameters into functions. This may allow you to create functions that modify the passed parameters rather than relying on return values.

1. Instead of random reels, model the reels as a rotating drum of symbols that spin (these can be numbers, letters or ASCII symbols) **–** you may be interested in the 'ncurses library', which provides more flexible text I/O functions than printf();

   Read the introduction to the ncurses library (sections 1-1.3): http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/intro.html, and see the 'helloworld' example. You'll need to `#include <ncurses.h>` in your C file and compile using '`-lncurses`' with gcc.

2. Display the reels as they spin by printing out the intermediate stages of the fruit machine;
3. Add additional game logic, e.g. hold and nudge functions (i.e. the user can select a reel to nudge by 1 position, or can select to hold up to n-1 reels);
4. Keep track of the winnings (score should increase proportionate to the number of reels matching) and output a frequency count of the number of reels matched in each win when the player quits the game.

You should be looking to make your code a) great to read, and b) solve the problem elegantly. Be sure to ask our opinion on your code in the labs.