ICS 212 Homework #4

In this homework assignment, add a claw crane machine to your gumball machine program. Note: I've never actually played one of these things before, so if you haven't either, you aren't alone. A long time ago, there was a hilarious Simpsons episode where Homer was trying to win a lobster harmonica, from one of these machines. The general idea with these machines is that you are supposed to maneuver a claw over a bin filled with prizes, and then you push a button, the claw drops, and it grasps the toy / prize, and then it drops the toy / prize into a chute for you to retrieve. The trick is that the claw is kind of clumsy and weak, and so it usually doesn't do a very good job of delivering the prizes (by design).

If you do a YouTube search on "claw arcade machine" you can see some videos of claw crane machines in action. I saw this video which was kind of good:

http://www.youtube.com/watch?v=Sio8QgKBOhU

● For this assignment, implement the bin of prizes as a three dimensional array.

● Implement a constructor, to create your "claw crane arcade machine". It should allocate a struct containing a three dimensional array.

● Create a function to fill the three dimensional array with prizes. Note that the level (i.e. the height at which the prizes are piled in the z direction) at each x,y coordinate does not have to be the same, and you should have some mechanism to have different kinds of prizes. So at a particular x,y coordinate, you might have a lobster harmonica, piled on top of a stuffed Angry Birds toy, on top of a Sponge Bob Square Pants toy. The prizes should obey the laws of physics, so that the pile of prizes should begin at z=0, and should be piled up sequentially, i.e. at z=1, z=2, z=3, etc. until there are no more prizes at a particular x,y coordinate (i.e. prizes can't hover in mid-air).

● You should also have a destructor function for your claw arcade machine.

●You should implement a function to print the contents of your three dimensional claw machine. A sensible way to do this would be to print a series of 2x2 'slices', i.e. at each z-level. A simple way to represent the contents of a coordinate in the 3-dimensional claw machine, would be to encode it as a single character, i.e. as a number or a letter. You should have a key, though, so that the user knows that L= lobster harmonica.

● You should also implement a print function which prints a grid representing the top of each stack at each x,y, because that is what is really relevant to game play.

● Create a function to implement the game play. This should be fairly simple. Ideally, we would make it so that the user would be able to control the claw with arrows or the mouse, but unfortunately C does not make this very easy. So, what I would recommend, is just to have the function provide the user with a range of possible X,Y coordinates, and then request desired X,Y coordinates from the user. Then the function should use a random number generator to (with some reasonable probability) determine whether the user has grabbed the prize, and you might want to print some messages and add some

delays to make it seem more exciting for the user. Of course, if the user wins a prize, you should also remove the prize from the machine (i.e. the 3D array). You should also have a mechanism for when a prize gets picked up by the claw, but then subsequently falls out of the claw, so that it gets moved from one column to another.

● Weave the claw crane machine into your program which includes gumball machines. The claw crane machine should reside in its own source code (.C) and header (.h) files.

● Replace the main loop of your program, so that it accommodates the multiplicity of machines. Previously this loop resided in function OperateGumBallMachine(), but now it should obviously be moved . You should probably like to have yet another module called arcade.c, and within this a function called ArcadeMainSelector() , or something along those lines. This function could contain the main loop, and could ask the user whether they want to play the claw crane machine, or whether they want to purchase gumballs, and if the latter, which gumball machine they wish to use.

●Observe that both the lobster claw machine, and the gumball machines will have some common functionality, namely they will both need to accept coins. Do you think they should keep this functionality separate, or do you think they should share a common function? They should share a common function shouldn't they? Which module should it reside in: ClawGameMachine.c, or GumBallMachine.c? Neither, do you agree, since the coin functionality would need to be called by both modules, so you would want to create yet another module called, Coins.c, and an accompanying header file called, Coins.h, or something along those lines.

As always, have fun!