The BinPacking is a program for playing a game called “Bin Packing”. In this game, the player is repeatedly presented with circular disks, of varying sizes, which he/she must place inside a rectangular bin. Each newly placed disk must be placed in such a way that it does not overlap any previously placed disks, and lies entirely within the rectangular bin.

The game program has a flaw. It only allows the user to place a limited number of disks in the bin. The problem is lies in the fact that the procedure \texttt{initialDiskArray} generates a limited number of disks and returns them in an array of disks. When all the disks in the array are placed, the game is over. (The program crashes.) Your task is to modify the program to remove this problem. Here is how you should do it:

1. Write a procedure to expand the size of the current disk array. The procedure should have the signature: \texttt{Disk[]} \texttt{expandedDiskArray(Disk[]} \texttt{oldDiskArray, Random randomizer)}. It should construct, fill and return a new array of disks that is twice as long as the input parameter \texttt{oldDiskArray}. This procedure should copy all of the disks from \texttt{oldDiskArray} into the first half of the new array. It should also create new disks to fill out the second half of the new array, using the same disk creation operations as found in the \texttt{initializeDiskArray} procedure.

2. Revise the \texttt{showNextDisk} method to detect when \texttt{diskArray} has been exhausted and replace \texttt{diskArray} with an expanded disk array.

After you have modified the BinPackking.java program, you should play the game a bit to make sure it is working properly. Do this by putting one or more break points at suitable locations in the program; running the program in debug mode; and verifying that \texttt{expandedDiskArray} is being called at the appropriate time. You should continue until the disk array has been expanded twice.

Now see how many disks you can place in the bin!