

Computer Science 1 – Grading Criteria
Program 1: *UCF KnightsBall Lottery*
Total: 100 Points (+ 10 possible bonus points)

Late Penalty: 10% for up to 24 hours late, and 25% for up to 48 hours late.

Following Development Guidelines (15 pts)

1. Used a reasonable struct to store information about one lottery ticket. (4 pts)
2. Used a dynamically allocated array to store all the ticket information. (7 pts)
3. Used an enumerated type to aid program readability. (4 pts)

Use of Algorithms (15 pts + 10 possible bonus points)

1. Has a correct algorithm to check the number of matches. (12 pts)
2. Properly matches the number of correct numbers with the winning amount (3 pts)
3. **Bonus points:* Uses the efficient $O(n)$ algorithm of comparing two sorted lists of size n for matches. (10 pts)

Here's the link to the PDF slides that shows the $O(n)$ algorithm:

http://www.cs.ucf.edu/courses/cop3502/sum2011/notes/COP3502_3_SortedListMatch.pdf

Execution Format Points (15 pts)

1. Opens the file properly (4 pts)
2. Properly reads in information from the file into the array of dynamically allocated structs. (4 pts)
3. Properly reads in the winning lotto combination (3 pts)
4. Properly writes to the output file (4 pts)

Execution Points (40 pts)

1. The input.txt file we use has 80 tickets. There are 10 winners***. So basically award 4 points per correct winner. Also, deduct 1 or two points per incorrectly stated winner (1 or 2 points depending on the severity of the error). *** Use winning combo: 4 12 29 32 36 46

Documentation and Style (15 pts)

1. Header comment with name, date, assignment info, etc. (4 pts)
2. Ample comments within the code. (5 pts)
3. Reasonable use of white space. (2 pts)
4. Reasonable indenting. (2 pts)
5. Reasonable variable names. (2 pts)

NOTES:

1. If the program does not use dynamically memory allocation for the array of structs, grade normally according to this criteria, and **then DEDUCT 70 points from that grade.** Meaning, at best, the final grade would be a 30, which is actually more than fair considering the write-up said no credit would be given to programs not using DMA.
2. If the program does not compile, spend, at most, five minutes to see if you can fix the compilation errors. If you can fix them quickly, grade using the above criteria and then deduct 30 points from the grade. If you cannot fix the compilation errors within five minutes, award a maximum of 50 out of a 100, but adjust the score based on what you can tell was done in the code.
3. If the program crashes, award at most 70 points based on what you see in the code.