

Lottery Simulation Program; Using Structures and Macros

Required Materials:

- Your textbook, *Assembly Language for x86 Processors* (7th edition)
- Removable or network device (Flash drive, memory card, MyMocsNet account mapped to a drive letter, etc.) for storage of your programs
- These instructions
- Intel-compatible, Windows-based personal computer (like the ones in EMCS 306) with text editor, MASM, and Visual Studio or CodeView (if needed for debugging)

Preparation for Laboratory:

Read the material on *structures and macros* in sections 10.1 and 10.2, pages 390-412 of your textbook. Also read the instructions below.

Instructions:

Write an assembly language program that will simulate a drawing of the Mega Millions lottery (see a description of how the game works and a list of prizes at <http://www.megamillions.com/how-to-play>. Your program must define and create an instance of a **structure** that represents a possible result of a draw (five white balls drawn from a drum containing numbers 1 through 70, plus one yellow Megaball from a second drum containing numbers 1 through 25). You must fill the structure with **five different randomly chosen numbers 1-70** plus **one randomly chosen number 1-25** (hint: use Irvine's Randomize and RandomRange procedures) to simulate the lottery draw. Display these numbers on the screen.

Your program must then prompt the user to enter the numbers from their lottery ticket (5 white and 1 yellow - store them in another instance of the same type of structure you used for the simulated draw) and check to see if the ticket is a winner. (See the list of winning combinations at the web site linked above to determine the prize amounts awarded; in short, if the user matches the yellow ball and/or three or more of the white balls, the ticket wins a prize.) **Note that the white balls can be matched in any order, not just the order in which they were drawn.** Your code must error-check the user input (e.g. do not allow the user to input duplicate numbers for the white balls, or any numbers that are out of range). If enough of the user's numbers match, output a message indicating the amount won (for matching all 6 numbers, simply output a message that the user wins the jackpot, since the actual jackpot amount varies). If there are an insufficient number of matches between the user's numbers and the lottery draw, output a message that the ticket is not a winner. (You are *not* required to simulate the Megaplier option; only the "normal" Mega Millions prizes paid on \$2 tickets.) After outputting the results, the program should exit. (Run it multiple times to make sure it generates a different lottery draw each time.) To reinforce the rest of the material covered in chapter 10, your program must also define and use at least one **macro** (additional use of macros is encouraged if you find them useful). Example program output follows:

```
Mega Millions drawing results: White balls 43 17 1 56 22 ... Yellow
ball 11
Please enter your first white number: 6
Please enter your second white number: 17
Please enter your third white number: 56
Please enter your fourth white number: 22
Please enter your fifth white number: 1
Please enter your yellow number: 6
You have matched 4 white numbers but not the yellow number. Your
ticket wins $100!
```

Payout Chart:

Match		Match	Prize *	Chances
5	+	1	Jackpot	1 in 302,575,350
5	+	0	\$1,000,000	1 in 12,607,306
4	+	1	\$10,000	1 in 931,001
4	+	0	\$500	1 in 38,792
3	+	1	\$200	1 in 14,547
3	+	0	\$10	1 in 606
2	+	1	\$10	1 in 693
1	+	1	\$4	1 in 89
0	+	1	\$2	1 in 37
Overall chances of winning any prize:				1 in 24

To Hand In: (due no later than 3:00 p.m. Tuesday, November 20)

1. Turn in a printed copy of the *thoroughly commented .LST file* for your program. **Be sure to follow the guidelines given in the programming style and documentation handout.**
2. Submit the *results* of your program as follows: print nine sample “screen shots” to show the output generated by the program for *each possible way to win* the lottery (matching only the yellow ball; matching the yellow ball plus one, two, three, four, or all five white balls; or matching only three, four, or all five white balls). Also print three more screen shots illustrating the *three losing possibilities* (no matches at all; or matching just one or two white balls but not the yellow ball).
3. Have the instructor check the operation of your program and sign in the space below when you have demonstrated its operation.

Instructor's signature for program demonstration: _____

Staple your program listing, results, and this signed sheet together. Submit these items by the date and time specified above. Late submissions will be penalized substantially.