Code Challenge

This is a quick code challenge to confirm your ability to make a dynamic webpage access a data source. We would like you to create a webpage that accepts 6 numbers from the user and checks those numbers against all of the BCLC's 6/49 lotto numbers since 1981, and reports back to the user an estimate of:

- dates they won more than $85
- how much they won in total (any prize amount)
- how much they spent in total on the 3,620 draws since 1981
- and the net win/loss in dollar amounts

You can download all of the historical 6/49 draws from this website:

https://lotto.bclc.com/winning-numbers/lotto-649-and-extra.html

Which will aim you at this zip file containing the CSV file:

https://www.bclc.com/DownloadableNumbers/CSV/649.zip

Sample of the data:

PRODUCT	DRAW NUMBER	SEQUENCE NUMBER	DRAW DATE	NUMBER DRAWN 1	NUMBER DRAWN 2	NUMBER DRAWN 3	NUMBER DRAWN 4	NUMBER DRAWN 5	NUMBER DRAWN 6	BONUS NUMBER
649	1	0	6/12/1982	3	11	12	14	41	43	13
649	2	0	6/19/1982	8	33	36	37	39	41	9
649	3	0	6/26/1982	1	6	23	24	27	39	34
649	4	0	7/3/1982	3	9	10	13	20	43	34
649	5	0	7/10/1982	5	14	21	31	34	47	45
649	6	0	7/17/1982	8	20	21	25	31	41	33
649	7	0	7/24/1982	18	25	28	33	36	42	7

1. Numbers entered by the user should be restricted to number between 1 and 49

2. The user must select 6 unique numbers, so check for duplicates

3. For this challenge, you can assume that tickets cost $1 per draw from draw number 1 to draw number 2124 in June 2004, then $2 per draw from draw 2125 to 2989 in September 2012, then $3 per draw after from draw 2990 to current draw 3620.

4. For this challenge, you can assume that prizes were the same for each draw, according to this table:

Numbers	Prize amount
6/6	$ 5,000,000.00
5/6+	$ 250,000.00
5/6	$ 3,000.00
4/6	$ 85.00
3/6	$ 10.00
2/6+	$ 5.00
2/6	$ 3.00

+ means bonus (5/6+ means 5 out of 6 numbers drawn, and the bonus matched)

Code Challenge

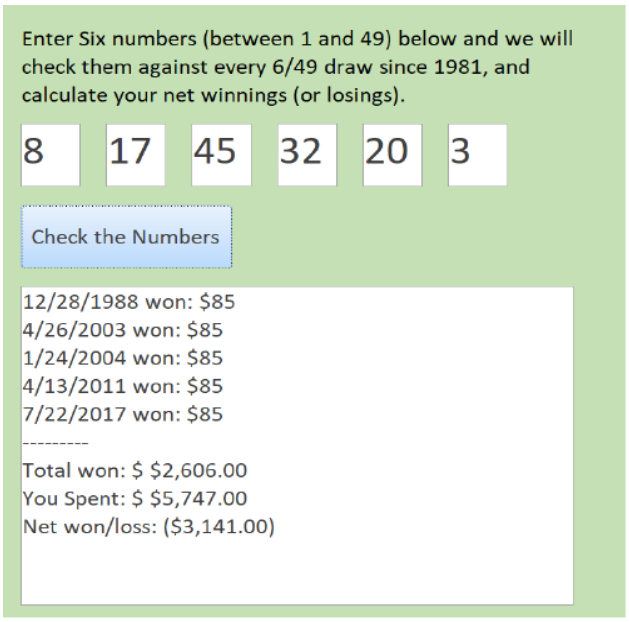
5. Only check against “Sequence Number” = 0 in the dataset. There are 74 draws where there are multiple sequences and I couldn’t determine why, so let’s leave those out.

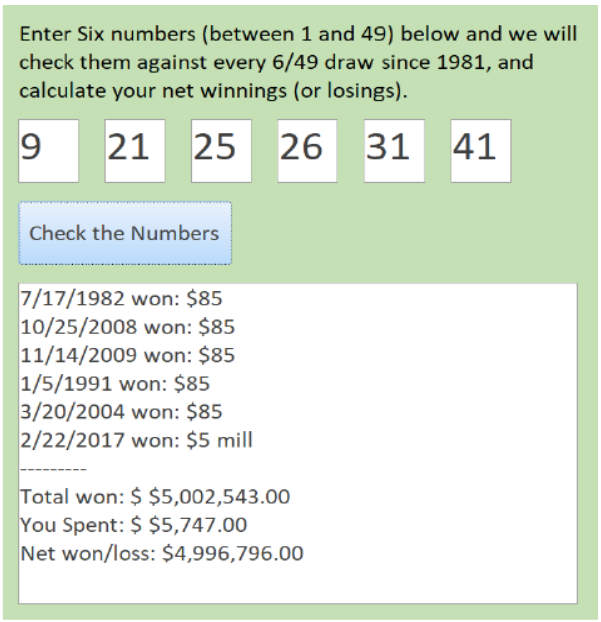
6. You do not have to have a mechanism to update the numbers for new 6/49 draws in the future, let’s assume draw 3620 is the last draw.

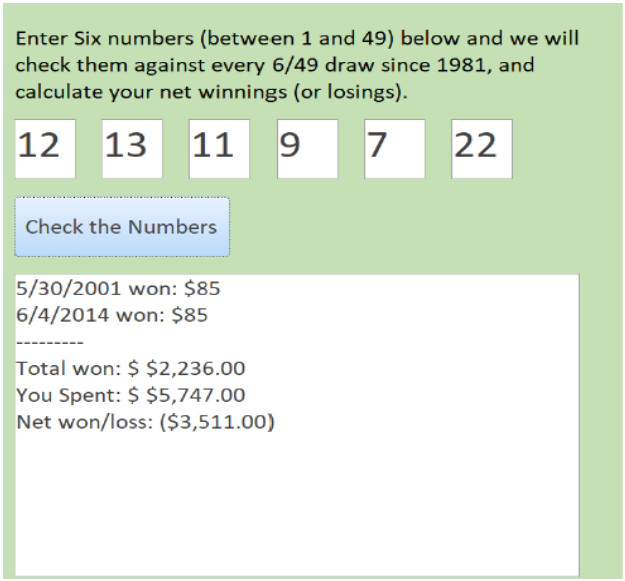
7. Your algorithm does not need to be fast, just accurate

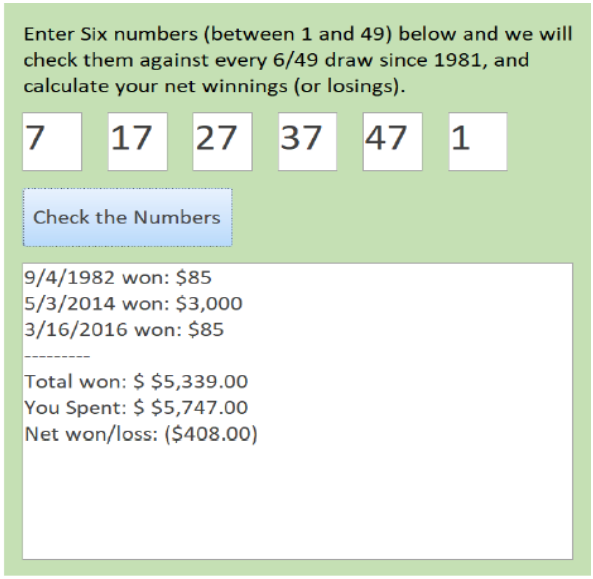
8. Only show prize dates for winnings $85 or more

Here are four examples of the webpage, numbers entered and the output:









Code Challenge

Other Notes:

You can build this web page anywhere, in whatever technology you like – we should just be able to view it publicly (even behind a password if you'd like) and it needs to be available for a week on the day of your submission. The code can be published to the provided repository for scoring. You will be provided an email address for questions, and all answers will be published for all teams to see.

Code Challenge Evaluation Rubric:

#	Criteria	Available Points
1	User interface: Can the user enter the numbers as specified?	15 points
2	Are the results of the query easy to read?	10 points
3	Does the page flow intuitively?	10 points
4	Was the code tested? And were the results accurate? (note: our results might not be correct)	15 points
5	Did the team write quality code (is the code logical, clear and instructive? Is the code properly documented? Does the code follow standard design patterns?	30 points
6	Did the team make good design decisions? Is the architecture well-organized and modular? Is the solution elegant and simple?	20 points
	Total:	**100 points**