Test Questionnaire

Role: Software Engineer

Rules

Solve problem No. 1 in within three days. You'll earn bonus points if you can solve problem No. 2 and/or No. 3.

For coding challenges (No. 1 and No. 2), provide results as:

A] Source Code in an online github account OR as a ZIP file

B] Deployed Example (for No. 1) online someplace (e.g. static files in S3)

C] Sample test.output (for No. 2)

Provide results inline in your email response or as a separate document for the SQL skills test (No. 3).

Solve this yourself without help from anyone else.

Feel free to reference online help - stackoverflow, google, etc.

No. 1 Tables Generator - Part 1

28	29			
27	26	25	24	23
18	19	20	21	22
17	16	15	14	13
8	9	10	11	12

Configure 20%

			247	246
241	242	243	244	245
240	239	238	237	236
231	232	233	234	235

Configure 30%

77	79	81		
75	73	71	69	67
57	59	61	63	65
55	53	51	49	47

Configure 40%

Table: **RED**

N = 8

X = 1

M = 29

W = 20 %

D = LTR-UP

OK CANCEL

No. 1 Tables Generator - Part 2

Rules

1. Build with HTML, CSS (or SASS, SCSS, LESS variants), ReactJS + Redux (or AngularJS)
   Provide results back via a github repo (for source) as well as a link to a deployed version of this application (to verify its runtime).
2. OK to use jQuery (if needed). Do not use Prototype, Backbone, Ember or any other JS lib or framework.
3. Start with a predefined state of the 3 tables rendered above.
4. Each table starts with a number N and increments by X from LTR then up, then RTL until a max M.
   Render void cells in gray.
5. Each colored div (red, green, blue) containing the table has a % width that resizes with the width of the window.
6. RESPONSIVE: If the window is resized smaller than a certain width, the blue table must disappear.
7. RESPONSIVE: If the window width is reduced further, than a certain width, the green table must wrap under the red table.
8. If the CONFIGURE button is clicked, a configuration panel appears below the rendered tables that facilitates modifications.
   OK applies changes and closes the panel.
   CANCEL discards changes and closes the panel.

Bonus Points

1. Client side stickiness; if I close my window and restore it, the state of the view is restored exactly as it was last viewed
2. Validate input values for data type, range
3. Obfuscate, minify, separation of css, js, html, etc; watch changes and rebuild
4. Unit test with Jest or Enzyme

No. 2 Number Generator

Expected output:

1. Generate a randomized dataset of 997940 numbers (containing values between 1 to 20)
2. This should contain the following distribution:

1 - 12	83,000
13	1,000
14	500
15	250
16	100
17	50
18	25
19	10
20	5

Clarification: For each of **1, 2, 3, …, 12**, there should be **83000** instances in the generated dataset.

3. No two consecutive numbers should be the same in the generated dataset
4. Write the generated dataset to a text file titled **test.output** with one value per line
5. On the console, display all lines on which the value 20 appears

Rules

1. Use Java as a programming language
2. You may reference any online documentation / help
3. Credit given on...
   a. Timely completion
   b. Accuracy of output
   c. Efficiency in programming
   d. Performance of execution
   e. Coding style and readability
   f. Documentation of source

No. 3 SQL Skills - Part 1

[A] Design and populate an associative table called **people_vehicle** to support the following:
- Mary has a Truck
- John has two Cars and two Trucks
- Chen has no vehicles

Share the contents of the **people_vehicle** table in your results in an email.

[B] Design and populate an associative table called **people_pet** to support the following:
- Mary has no pets
- John has a Dog
- Chen has a Parrot and a Cat

Share the contents of the **people_pet** table in your results in an email.

people

id	name
1	John
2	Mary
3	Chen

vehicle

id	name
1	Truck
2	Car
3	Bike

pet

id	name
1	Dog
2	Cat
3	Parrot

No. 3 SQL - Part 2

NOTE: Feel free to use PostgreSQL, MySQL or Oracle specific dialects for [C]

[C] Write a query that generates exactly this result set:

name	vehicles	pets
John	Car=2, Truck=2	Dog=1
Mary	Truck=1	None
Chen	None	Parrot=1, Cat=1

[D] Write a query that finds all people who have at least 2 vehicles or 2 pets. Share the query in your response.

people

id	name
1	John
2	Mary
3	Chen

vehicle

id	name
1	Truck
2	Car
3	Bike

pet

id	name
1	Dog
2	Cat
3	Parrot