This is a technical screening to set a bar for candidates. We will use the results of these questions, your resume, and any additional information you send us to determine whether we think a longer conversation will be worth yours and our time.

Feel free to answer these in a separate file or document and send that to us or send it as a series of individual files, whatever makes the most sense to you. Typically this takes less than 2 hours; one person completed all required and optional questions in under an hour (with unit tests).

Section 1: (complete all 3 of these)

**1. (C#)** Write a function that accepts two strings and produces an interleaved result composed of alternating letters from each source string. Example, given "abc" and "123", the result would be "a1b2c3".

**2. (C#)** Write a method that takes two strings and calculates how many times the second string exists in the first. Example, given "redgreenbluered" and "red", the answer would be 2.

**3. (SQL)** We're writing a "Current Open Tickets" report for a ticketing system that tracks Tickets and their Status changes in two separate tables (for auditability). Write a SQL statement using the two tables below that returns every open ticket (latest status is not Closed) from the "Ticket" table with its current status (status with the latest timestamp) from the "StatusChange" table

Ticket Table

| **Id** | **Summary** |
| --- | --- |
| 1 | "My Computer won't turn on" |
| 2 | "Could you print the date out in my timezone on Report XYZ?" |
| 3 | "Could you create a new version of the XYZ report that has 2 new columns?" |
| 4 | "Where is the password reset link?" |

StatusChange Table

| **TicketId** | **Timestamp** | **OldStatus** | **NewStatus** |
| --- | --- | --- | --- |
| 1 | 2014-01-01 00:00 | NULL | "New" |
| 1 | 2014-01-01 01:15 | "New" | "In Progress" |
| 2 | 2014-01-01 02:32 | NULL | "New" |
| 3 | 2014-01-01 04:53 | NULL | "New" |
| 2 | 2014-01-01 05:07 | "New" | "In Progress" |
| 2 | 2014-01-01 06:14 | "In Progress" | "Closed" |
| 4 | 2014-01-01 07:22 | NULL | "New" |
| 1 | 2014-01-01 08:25 | "In Progress" | "Closed" |
| 2 | 2014-01-01 09:32 | "Closed" | "Reopened" |
| 3 | 2014-01-01 10:55 | "New" | "In Progress" |
| 4 | 2014-01-01 11:05 | "New" | "Closed" |

Expected Result:

|  |  |  |
| --- | --- | --- |
| **Id** | **Summary** | **Status** |
| 2 | "Could you print the date out in my timezone on Report XYZ?" | Reopened |
| 3 | "Could you create a new version of the XYZ report that has 2 new columns?" | In Progress |

Section 2:(complete one of these)

**4A: (javascript)** Create a snippet of HTML and javascript to display a column chart of the values below. Feel free to use external charting libraries.

var rawData = [  
/\* year, sales, expenses \*/  
['2004', 1000, 400],  
['2005', 1170, 460],  
['2006', 660, 1120],  
['2007', 1030, 540]  
];

**4B: (SQL)** Using the sample data tables from #3, calculate the total amount of time each ticket spent in each available status. The result is expected to have a row for each ticket row in the Ticket Table and a column for each potential status (New, In Progress, Closed, Reopened), with values for the number of minutes spent in that status.

For comparison, if you run the report with the current date of '2014-01-01 12:00' (ie, use this date as the end date for the last new status for each ticket), you should receive results like:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Summary** | **New** | **In Progress** | **Closed** | **Reopened** |
| 1 | "My Computer won't turn on" | 75 | 430 | 215 | NULL |
| 2 | "Could you print the date out in my timezone on Report XYZ?" | 155 | 67 | 198 | 148 |
| 3 | "Could you create a new version of the XYZ report that has 2 new columns?" | 362 | 65 | NULL | NULL |
| 4 | "Where is the password reset link?" | 223 | NULL | 55 | NULL |

**4C: (C#)** Write a function that takes a source directory path, a search string, and a destination filename. The function should then open all of the files in the given directory in parallel (using parallization technique of your choice), find lines that have the search text in them, and extract and output all lines with that search text in a file with the given destination filename. At the end of execution, the function should output the number of files it processed, the number of lines it found the search text in, and the number of occurrences of that search term (don't assume once per line).