Campaign Monitor
by MARIGOLD

Coding Take-home Test Questions

Introduction

As a developer with Campaign Monitor, a very large part of your role will be writing code. So, as part of the recruitment process, we want to give you an opportunity to show off your skills.

The code review involves solving a set of questions which are outlined below. They are designed for you to work on at home in your own time, meaning there's no 'deadline'.

We expect most applicants to spend roughly 90-240 minutes working on these questions.

Once complete, please forward us a zip of the source code and dependencies the inbox referenced in the email*. You may wish to rename the .zip file to .piz to keep your mail server happy.* Rather than a zip file you may, if you prefer, send a tar file or git bundle file instead.

If your mail server blocks the attachment, feel free to use Dropbox etc to share the file with us.

You should use whatever language you are best at and select the questions which are best suited to showing off your skills and experience. It's recommended to stick to the suggested languages for each question but you are free to implement your answer in a language that is not listed if you so choose.

You do not need to complete all the questions. The questions are labelled with points according to the expected time investment. **You must complete a minimum of 6 points worth of questions.**

If you use online resources to assist with solving a requirement, please include the url (in a comment in the code).

Your solution will be evaluated internally by one or more of your potential co-workers. You should expect a response from us within 4-6 business days (please feel free to follow-up if we don't get back to you within 6 business days). We will let you know if you have proceeded to the next stage, and will also (optionally) provide an evaluation, as feedback, on your code review application.

Guidelines for Submissions

When writing your code, please be mindful of the following:
Your code should be free of bugs, and solve the stated problem.
Your code should be easy to understand and maintain by other developers.
Your code should demonstrate consideration for handling common error scenarios.
Unit tests are expected where relevant (feel free to use your preferred unit test framework).
If you have made any assumptions in solving the questions, please mention those assumptions in the comments in the code.

.NET/C#

If you're using .NET, we expect the questions to be completed using C# (preferably using Visual Studio 2022 or newer, or Visual Studio Code). You may also use other libraries, NuGet packages, unit testing frameworks, etc. – but please ensure all dependencies are included in the file you send to us.

Java/Scala/Kotlin

Please include a build file explicitly listing all project dependencies.

For Java/Kotlin projects, it is preferable to use Gradle/Maven - include source and target JDK versions. For Scala projects, it is preferable to use SBT - include Scala version used.

Python

Please add a README file which makes it clear which version of Python you've used (we prefer Python 3.5 and above but will accept Python 2.7 or higher) and please include a requirements.txt file to list any pypi dependencies you have used in your solution as well as basic instructions for running your solutions and tests.

Javascript

Please add a package.json with any dependencies/scripts set up to run and test the code.

SQL

The questions were written with T-SQL (Microsoft SQL Server) in mind, but you are free to submit answers in SQL for SQLServer, Postgres or MySQL. (We use multiple different SQL databases at Campaign Monitor).

Please add a README file which makes it clear which database and version you've tested your SQL on and try to avoid database-specific features where possible.

Other languages

If you would prefer to submit the test in an alternate language to those listed above, please:

1) Inform your contact at Campaign Monitor or the recruiter that you will do so, and advise what language it will be to ensure it can be reviewed.
2) Please include instructions for installing any necessary dependencies and running your solutions.

Question 1 (1 point, any language)

Using the most appropriate means, implement an "isNullOrEmpty" check on the standard String type. It should be functionally equivalent without calling any "isNullOrEmpty" built in function.

Sample Inputs	Sample Outputs
null	true
“a”	false
“”	true
“null”	false

Question 2 (1 point, any language)

Write a function that takes a single positive integer, and returns a collection / sequence (e.g. array) of integers. The return value should contain those integers that are positive divisors of the input integer.

Sample Inputs	Sample Outputs
60	{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60}
42	{1, 2, 3, 6, 7, 14, 21, 42}

Question 3 (1 point, any language)

Write a function that takes three integer inputs and returns a single output. The inputs are the lengths of the sides of a triangle. The output is the area of that triangle.

Sample Inputs	Sample Outputs
3,4,5	6
Any inputs that are negative	(throw) InvalidTriangleException
Inputs that cannot form a valid triangle	(throw) InvalidTriangleException

Question 4 (1 point)

Write a function that takes an array of integers, and returns an array of integers. The return value should contain those integers which are most common in the input array.

Sample Inputs	Sample Outputs
{5, 4, 3, 2, 4, 5, 1, 6, 1, 2, 5, 4}	{5, 4} *
{1, 2, 3, 4, 5, 1, 6, 7}	{1}
{1, 2, 3, 4, 5, 6, 7}	{1, 2, 3, 4, 5, 6, 7}

** note that order of the elements in the return array is not significant - so {5, 4} or {4, 5} is fine*

SQL Questions

The following questions are based on the tables presented below. *Your answers should be entered into a file called **SQL.txt.***

Salesperson

SalespersonID	Name	Age	Salary
1	Alice	61	140000
2	Bob	34	44000
6	Chris	34	40000
8	Derek	41	52000
11	Emmit	57	115000
16	Fred	38	38000

Customer

CustomerID	Name
4	George
6	Harry
7	Ingrid
11	Jerry

Orders

OrderID	OrderDate	CustomerID	SalespersonID	NumberOfUnits	CostOfUnit
3	17/01/2013	4	2	4	400
6	07/02/2013	4	1	1	600
10	04/03/2013	7	6	2	300
17	15/03/2013	6	1	5	300
25	19/04/2013	11	11	7	300
34	22/04/2013	11	11	100	26
57	12/07/2013	7	11	14	11

An example question and answer is:

Example Question	Example Answer
Return the name of the customer with the largest value of CustomerID	SELECT TOP 1 [Name] FROM Customer ORDER BY [CustomerID] DESC

Question 5 (1 point, SQL)

Referring to the above schema, please write SQL queries to do the following:

A. Return the names of all salespeople that have an order with George
B. Return the names of all salespeople that do not have any order with George
C. Return the names of salespeople that have 2 or more orders.

Question 6 (1 point, SQL)

Referring to the above schema, please write SQL queries to do the following:

A. Return the name of the salesperson with the 3rd highest salary.
B. Create a new roll-up table BigOrders(where columns are CustomerID, TotalOrderValue), and insert into that table customers whose total Amount across all orders is greater than 1000
C. Return the total Amount of orders for each month, ordered by year, then month (both in descending order)

Question 7 (2 points, Javascript)

Write a function which combines an array of objects, grouped by a key you provide (this key will correspond to a key found in the objects. The function will index the new object with the value of those keys.

Sample Input array	Sample Output object (using 'name' as key)

const users = [{
id: 1,
name: 'bob',
},
{
id: 2,
name: 'sally',
},
{
id: 3,
name: 'bob',
age: 30,
}];
	{
	bob: [{
	id: 1,
	name: 'bob',
	},
	{
	id: 3,
	name: 'bob',
	age: 30,
	}
	],
	sally: [{
	id: 2,
	name: 'sally',
	}],
	}

Requirements
The function must not mutate the original array, or any of the objects in the array.
If the key you provided is not present in an object, exclude that from the output.
If an element of the array is null/undefined, or not an object - exclude that from the output.
It should handle one or more objects being attached to a key.

Finally, write the function so it can store the key, and later be invoked with an array.
 	
Example:
const arrangeByName = arrangeBy('name');
arrangeByName(users);

Please provide unit tests which prove this functionality.

You can write this exercise using modern Javascript available to browsers today (Chrome, FF, Safari, Edge).

Question 8 (3 points, any language)

Link Checker

You have a large bunch of HTML. Inside that HTML are p tags, li tags, table tags, really any and all kinds of HTML tags. Most importantly there are anchor/link tags.

Write a program to find all of the URLs to which those link tags link and verify that the URLs return a 200 response. In a given chunk of this HTML, we could have anywhere from 0 to 100+ links, so your solution should handle the case where there are plenty of links.

Requirements
First, you'll want to figure out a way to extract all of the URLs.
Second, you'll want to test the URLs and report back to the user which are valid and which are not.
Third, you'll want to make it really fast by checking the URLs concurrently or by parallelizing the checks. You might want to think about caching as well.