Backend/Server Programmer

A. ALGORITHMIC AND DATA STRUCTURES

1. [Q1] The expression (a && b) || c evaluates the same as:

○ (c || a) && (a && b)

● (c || a) && (c || b)

○ (c && a) || (c && b)

2. [Q2] The expression ((a || !a) || false) || ((a && true) && (a && !a)) evaluates as:

○ !a

○ false

● true

○ a

3. [Q3] When would you use a linked list over a vector? (There may be more than one correct answer)

□ Always

□ For fast search

☑ For fast insertion

□ For fast iteration

4. [Q4] Describe two containers. Give their complexity to

I am going to briefly describe Arrays and Linked Lists. Each of them has their advantages and disadvantages and the choice must be made according to the usage circumstances, as with most choices in computer programming and design.

Array: allows random access to any element, so looking for specific elements can be pretty fast (which means that if you know at what index you item is, you can retrieve it with complexity O(1)). Has fixed allocation size, which means it might have to re-allocate the memory on a new array with double the capacity if adding a new item beyond the capacity of the array, which means a worst case of write of O(n).

Linked list: doesn't allow for random access, which means that the complexity for retrieving an element is O(n), but adding an element to it is O(1). It is linear and the elements are sparsed in the memory, so the order is defined by each element, which points to the next (or the previous in the case of double linked lists). It does not require reallocation when adding new elements, contrary to a basic array.

a) Insert

Array:

Can be O(1) for a Specific element, but usually is O(n) if re-allocation is necessary.

Linked list:

O(1) in the beginning;

O(1) in the end when last element is known or O(n) when last element is unknown;

O(n) in other cases (which includes searching for the element)

b) Search

Array: O(1) if you know the index, otherwise O(n)

Linked list: O(n)

c) Count

Array: O(1) because of the fixed size.

Linked list: O(n) if no counter is kept, because you need to traverse each element.

5. [Q5] The Fibonacci sequence is defined by f(n) = f(n-1) + f(n-2), with f(0) = 0 and f(1) = 1. The first iterations are: 0 1 1 2 3 5 8 13… In the language of your choice (preferably C# or C++), design a recursive algorithm that returns the Nth Fibonacci number.
For instance, f(6) = 8.

```
#include<iostream>

using namespace std;

int fibonacci(int n)
{
    // base case. Can be read as if(n<=1):
    if((n==1)||(n==0))
    {
        return(n);
    }
    else
    {
        // recursive call:
        return (fibonacci(n-1) + fibonacci(n-2));
    }
}

int main()
{
    int n;
    cout<<"This is a recursive algorithm that returns the Nth Fibonacci number.\n";
    cout<<"Please input N:";
    cin>>n;

    cout<<fibonacci(n);

    return 0;
}
```

6. [Q6] What is the time complexity of the recursive algorithm you design for the Fibonacci sequence of the algorithm?

○ O(n^2)

○ O(n)

○ O(logn)

● O(2^n). *Recursion is nice and cool, but usually not the best option for optimization!*

7. [Q7] If your Fibonacci algorithm was design sequentially what would the time complexity be?

○ O(logn)

● O(n)

○ O(n^2)

○ O(2^n)

8. [Q8] Describe one sorting algorithm you know in a few words. What is its complexity?

MergeSort:

1. divides an input array in two halves
2. calls itself for the two halves, until their size is 1
3. merges the two sorted halves

It is a Divide and Conquer algorithm with complexity O(n log n) it all cases (best, average and worst). It is not the fastest, nor the slowest for every context, but it has a reliably consistent performance, which makes it a really good general-purpose option.

9. [Q9] Which of these sorting algorithms provide the best average solving time for most cases?

○ Bubble Sort

○ Insertion Sort

● Quick Sort

B. ARCHITECTURE AND DESIGN PATTERNS

1. [Q1] What are design patterns and why are they used in software engineering?

They are great to solve a lot of issues that arise on software engineering. Specially in larger projects, code can get very messy as thins evolve naturally, so instead of just coding it is a good idea to have a pattern to follow when designing software.

Different names can end up being used for the same things, or their difference is not clear. Code written and maintained by many people can become too complicated too easily if everyone is following their own styles, or developing their own coding patterns, until nothing makes sense anymore. Controller, handler, executer, provider, manager, logic, orchestrator and coordinator are just some of the names that can be found around the same project.

So Design patterns can tackle these kinds of problems. They can be seen as a common basis for the people working on a project in order to be organized when coding and also help manage the complexity of projects. They can also make the solution of a problem super easy and intuitive, like using memento to implement the functionality of go back or go forward on a text editor.

Some people focus too much on the algorithms part of development, but in the real world, design patterns are much more useful. By this I mean it is nice to know how to implement a merge sort function, but in reality, nobody implements that, they just use it! As for design patterns, it is great to know about them so you can think more broadly about the design of a system and to make actually good design decisions that shape a project into something different than an over-engineered garbage that is too hard to be maintained.

2. [Q2] Which design pattern is self instantiating and provides a global point of access?

○ Abstract Factory

● Singleton

○ Visitor

○ Object Pool

3. [Q3] Which pattern allows a class to change its behaviour at runtime by defining a family of algorithms, encapsulating each one, and making them interchangeable?

● Strategy

○ Composite

○ Facade

○ Visitor

4. [Q4] What are the disadvantages of the singleton pattern?

This design pattern is basically where you have globals flooding the system. Singletons are great, kind of because they exist in their little area and they don’t have lot’s of dependencies. It works well for many teams because each team can create their own singletons and it doesn’t effect anybody else. However, they can be pretty hard to test, but you can also use dependency injection to scope the singletons a little bit more and to make them more testable.

Another problem with them is that they can easily introduce two-way data-flows in your programs. For example, you have singleton A and singleton B talking to each other and the data flow is just two way. Or if you have tens or even hundreds of singletons, they can all be messaging each other, you would have a huge data flow, and if you had to trace the data flow you would have to go from one singleton to another, to another, to another and so forth. It would extremely hard to understand how the whole system is working. Another pattern can be used to solve this two-way data flow problem, if you had a central notification hub that would take care of the communication with the singletons, for example, like the publish-subscribe pattern.

There are other problems, like having bugs if the state of the singleton changes unexpectedly, since the state is shared globally.

5. [Q5] What is the design pattern used in this snippet of code?

```
class Shape
{
    public virtual void Display() { }
}

class MultiShape : Shape
{
    private List m_Shapes = new List();

    public void Add(Shape shape)
    {
        m_Shapes.Add(shape);
    }

    public void Remove(Shape shape)
    {
        m_Shapes.Remove(shape);
    }

    public override void Display()
    {
        foreach (Shape shape in m_Shapes)
        {
            shape.Display();
        }
    }
}

class Ellipse : Shape
{
    public override void Display()
    {
        Console.WriteLine("Ellipse");
    }
}


class Rectangle : Shape
{
    public override void Display()
    {
        Console.WriteLine("Rectangle");
    }
}

// Program entry point
public class Main
{
    private int main()
    {
        Ellipse ellipse1 = new Ellipse();
        Ellipse ellipse2 = new Ellipse();
        Rectangle rectangle1 = new Rectangle();
        Rectangle rectangle2 = new Rectangle();

        MultiShape multishape1 = new MultiShape();
        MultiShape multishape2 = new MultiShape();
        MultiShape multishape3 = new MultiShape();

        multishape2.Add(ellipse1);
        multishape2.Add(ellipse2);
        multishape2.Add(rectangle1);

        multishape3.Add(rectangle2);

        multishape1.Add(multishape2);
        multishape1.Add(multishape3);

        multishape1.Display();
    }
}
```

Looks like it is the Adapter pattern, commonly used in C# development.

6. [Q6] In a FPS game, we have a class to represent bullets: Bullet. We know we're going to create and destroy a lot of instances of this class. One strategy is to allocate dynamically a new instance of the class every time we need a new bullet. What are the drawbacks of such a strategy? What strategy would you use to mitigate those risks?

I'm not sure. I guess we would have many duplicated data flowing around and we could also reach a limit in memory.

On the other hand, if we had an instance of this class we could make copies of it, reducing the overhead of creating new instances and also tweak the copy constructor.

I don't like that everything related to a bullet would have to be created and then destroyed all the time. Maybe a static variable, or even a singleton could be used in this case, so that common shared attributes stay in this object, while we would be able to spawn news bullets that shared memory with this object.

7. [Q7] Explain the following subjects: What they are, how they work, one example of when would you use it.

i. Reflection

It's a feature of programing languages, it works by allowing the language to be it's own metalanguage or, in other words, code that can modify it's own behaviour. This is very useful for dynamic situations in which you need the code to adapt.

ii. Resource acquisition is initialization

Another feature of languages, that handles the lifecycle of a resource so it avoids exceptions or deadlocks. It works by assuring that a resource will exist in a scope until it is released by a destructor. It's useful when exception-safety is important. It's very useful in multithreading scenarios.

iii. Inversion of control / Dependency inversion

I am just learning about this. From what I understood, dependency inversion is a programming principle, that works by making sure that high-level modules depend on abstractions rather than concrete implementations of lower-level modules. Inversion of control is an implementation of dependecy inversion. It is useful for achieving loosely coupled code, making it easier to make modifications to code without afecting dependencies.

D. MULTITHREADING

1. [Q1] Choose the correct statement:

○ Process-based multitasking allows computers to run two or more programs concurrently

○ Programs act as a small unit of code that can be dispatched by the scheduler

● Both

2. [Q2] Choose the statements which differs the thread based multitasking and process based multitasking from each other. (There may be more than one correct answer)

☑ Process-based multitasking handles the concurrent execution of programs

□ Process-based multitasking handles the concurrent execution of pieces of the same program

□ Thread-based multitasking handles the concurrent execution of programs

☑ Thread-based multitasking deals with the concurrent execution of pieces of the same program

3. [Q3] How can you force shared resource synchronization? (There may be more than one correct answer)

☑ Using a semaphore

□ Using a thread

☑ Using a mutex

☑ Using a spinlock

4. [Q4] Select the correct statements in the following: (There may be more than one correct answer)

□ Coroutines generalize subroutines for preemptive multitasking

☑ Coroutines are well suited to implement iterators

□ A coroutine execution might be interrupted by another coroutine

□ A coroutine execution might be suspended by another thread

E. BACKEND ARCHITECTURE

1. [Q01] In Object-Oriented Programming, a lot of people state that Composition is often better option than Inheritance. What do you think about that?

Abstraction is at the heart of problem solving, it has to do with the representation of reality.

When solving problems with computers, it is necessary to represent the real world in code. One way of doing this is through the Object-Oriented paradigm, which can be seen as a simple set of tools to abstract a problem and it's.

However, an abstraction is never the same as reality, it can only be so accurate as the modeling. In this sense, it is interesting to use whatever tools are available at our disposal, which means both Composition and Inheritance should be used. Both might be used to represent the very same thing and make a software work, however a solution that is not optimal could come with many problems.

So, if we are modeling cars in software, it doesn't make sense to have a relation of inheritance between a car and an engine, because a car is not an engine, it has an engine. The car integrates an engine, interfacing with the engine to compose a higher level of abstraction, or a more complex thing, and provide more value to their users.

All in all, Composition can be a better option than Inheritance, and vice versa, depending on the problem being tackled and the orientation of the solution.

2. [Q02] Can you explain the difference between SQL vs NoSQL Databases

No. I mean… Yes! SQL is SQL and NoSQL is not SQL!

OK, let me clarify: SQL databases are relational, which means the data it contains is related, one entry to another. The data is defined and manipulated based on the Structured Query Language, which is not present in NoSQL databases. In the case of NoSQL databases, the data is just there, stored in an unstructured way, which means it can be anything. Both have advantages and disadvantages between them, and the choice of which one to use for a project must be made according to the usage circumstances, as with most choices in software design. A combination of both can also be used, to take advantage of both worlds.

The differences extend beyond their definitions: performance, scalability, practicability of usage, the "CRUD" syntax, how transactions are executed, how data integrity is handled, normalization, how to relate the data and maybe other aspects I didn't remember now.

I would say that the most important factors when choosing which one to use are:

Logical and data requirements are clear? Data integrity is super important? SQL!

Dealing with unrelated data, unclear or evolving requirements? Speed and scalability are super important? NoSQL!

NoSQL databases are definitely trending nowadays, with MongoDB, Redis, Cassandra and others. It is so trendy that even SQL databases have support to be used as NoSQL, for example PostreSQL's JSON field, which can actually make the database able to take advantages of both worlds.

3. [Q03] When would you choose a NoSQL database over a traditional Relational Database Management system?

It depends on the requirements, like the kind of data that they are going to handle. Both have advantages and disadvantages between them, and the choice of which one to use for a project must be made according to the usage circumstances, as with most choices in software design. A combination of both can also be used, to take advantage of both worlds.

I would say that the most important factors when choosing which one to use are:

Logical and data requirements are clear? Data integrity is super important? SQL!

Dealing with unrelated data, unclear or evolving requirements? Speed and scalability are super important? NoSQL!

4. [Q04] Assuming you have a Relational database system that's already in production. The current setup is near it's limits and we need to scale it to make sure it can handle additional load. What are possible techniques to achieve this goal?

I don't know what do you mean by the limit of the current setup. What is the bottleneck? Bandwidth? Access times? Insertion times? Storage capacity? Processing power?

Anyway, a simple solution is always throwing money at the problem, which in this case, would mean using better hardware (more CPU cores, more disk space, faster disks, more RAM, faster RAM, closer proxies, bigger bandwidth, etc).

A more thoughtful solution could be the normalization of the database entities, migration of legacy data, adding indexing to important tables, tweaking the configuration of the server.

Combining these solutions is even better, because just adding more hardware is not optimal. Having the database replicated in more than one server (master who handles writes and slaves that read), with a load balancer deciding which one should reply to a request also demands that you set up all of these servers correctly, so this should also require scripts to automatically set them up.

A revision of the architecture could even be necessary. So, instead of having a monolithic database system with a load balancer, a better approach could be separating the database for each micro-service or hosts while all could connect to a central authentication server. This can be simpler, because a load balancer can be hard to manage, set up and it also costs extra money.

Another possibility is to use a caching layer, that can be implemented by something like Redis, that would reduce load on the database server because it could take care of heavy queries or frequent queries.

Finally, what I can think of is to optimize the SQL queries being run. Keeping them simple, doing joins on the logical code instead of directly on the database, or running cron jobs to pre-calculate things and then insert them on the database, instead of making these calculations in the database server.

5. [Q05] You have a web API server that is having performance issues. Describe the procedure you would use to find the source(s) for this issues? If you can name tools that you have used to perform this type of analysis.

Many tools can be used to find out the cause of performance issues. Some are lower level, like using siege on a console to send many requests to the server. Some are more advanced, like nmap, to crawl through all exposed endpoints of a server. Others are very high level, like the many online tools that pop up if you search for them on DuckDuckGo, in the end they are just a collection of lower level tools that can give an idea of where the problem is, but won't really replace the criteria of someone looking for the root causes.

Tools are just tools, it is important to find out where the problem is. It could be that the web server need more configuration (like nginx, tomcat or apache). The server could be under attack. An endpoint could be running code that is not optimized. The database in which the API connects to could be having performance issues that propagate to the API. The user base has increased and the server just needs more hardware. Etc, etc, etc. There are so many reasons for an API server be having performance issues that solving the problem is a matter of brute forcing through all the data available, in form of logs or system resource analysis, and creating tests to reproduce the problem. Intuition can play a big role in this, after all, since it can be anything, it doesn't matter what is going to be tested first.

The first step is definitely paying attention to what is said by whomever reported the issue with the server, or what an alert is for.

The second step is to see what is the bottleneck that is being hit.

After that, it's definitely interesting to see if it is only a specific endpoint that is causing problems, but then how I would proceed to find what is the problem would depend on the clues I would have collected so far.

6. [Q06] Describe the differences between Synchronous and Asynchronous processing, what are the benefits and drawbacks of each approach. When would you choose to use a synchronous process vs and async one.

In synchronous processing happens as they come, in order. Asynchronous, on the other hand, is the contrary, there is no order, and processing can wait. It's like the difference between talking to somebody on the phone when both parties are available or leaving a message when only one of the parties is available.

The first can be more expensive, while the latter cannot guarantee when the results will be ready.

The choice usually comes down to: if you can wait, do it asynchronously, because it can process in batches, for example, and it will be cheaper. But if time is an issue, process it synchronously

7. [Q07] You are tasked to ensure that the current solution and architecture meets the quality, scale and availability levels required. Which kinds of tests would you implement along the process and what is the goal of each test type?

Quality can include so many non-objective things which can make it a little harder to test. The code should be written in a way that makes it easy to maintain and evolve, so code revision is a must to test for quality. The code should also do what it should be doing, so writing codes for it is a way out, but the specification must be clear. The configuration of the servers is also subjective, and needs inspection. The high level design choices are also subjective and the test is basically checking the documentation along side the requirements. Quality includes even more concepts, like how the system behaves in case of failures? Are there backups? Where are they? What is the downtime if we need to restore a backup? What is the host of the infrastructure? Does the architecture can easily accommodate new entities or changes? What would happen if something needed migration? How are updates being delivered to the servers? What risks were taking during the design? Etc. So in short, testing for quality is a subjective process that needs criteria from someone. I would also say that ideally more than one person should analyse the quality of an infrastructure. They should be people with broad knowledge.

For scale it is easier to test, because all that is needed is to mock heavy load and see how the system behaves. A sandbox or testing environment that is identical to a production environment should be used for this. The tests can be automatized, which is nice, and all they need to do is to be sure they hit every endpoint. There are basically two types of test to be implemented here, one that mimics how an organic traffic would look like and another that puts the system to full load.

To test for availability basically we need to ask the question: what happens with the system if there is a failure? So to this test can also be automatized. It is about simulating failure, to find out what is the downtime in case a backup need to be restored, or a rollback applied to the system's platform? Usually solutions to availability are limited by costs, because it's not free to keep a copy of your system to bring it up whenever there is a problem with the main system, for example. So it is also important to consider what risks when taken.

It is also a good idea to test for robustness of a system, which means trying weird input to see how it behaves.

8. [Q08] The Brewer's Theorem, most commonly known as the CAP theorem, states that in the presence of a Network Partition (the P in CAP), a system's designer has to choose between Consistency (the C in CAP) and Availability (the A in CAP). Can you think about examples of CP, AP and CA systems?

In case of a system failure…

CP: No availability guaranteed, which means upon finding a failure the whole system must wait, while the system remain unavailable until the data is consistent again and the fix is propagated.

AP: No Consistency guaranteed, which means a system can respond with whatever data is available, but it doesn't necessarily mean it's the latest value for that data.

CA: No Partition tolerance, which can be a system that implements a master/slaves architecture. The master becomes a critical point of failure and there are also scaling limitations for this.

9. [Q09] Scale out vs scale up: how are they different? When to apply one, when the other?

Scale up is getting better hardware for a server hosted by a single unit of hardware.

Scale out is fragmenting a server in more than one unit of hardware.

Going up can be more expensive, but also is easier to maintain and doesn't require modifications to a deployed system.

Going out can be cheaper in terms of hardware cost, but might need refactoring of an architecture and can be harder to maintain.

10. [Q10] Why do we usually put so much effort for having stateless services? What's so good in stateless code and why and when statefulness is bad?

People can make jokes about how funny it is to reboot a machine or a system when it crashes, so after restarting it behaves normally. This is not magic, it just happens that when you reboot a system it starts behaving from an initial state, and before it could have been in an unknown or unexpected state in which it didn't know how to behave, or could not get out of, like a deadlock. So we put effort for having stateless services because it greatly reduces the frequency of these kinds of problems, of when a system is in an unexpected state.

Making a reference to a state can also be difficult in a stateful system, which makes testing harder, for example, because you would have to mock a lot of pre-states before getting to the one you really want to investigate. In other words, these systems are highly dependant on the user behaviour, which can be unpredictable, and it's harder to make them robust.

Stateless code is good because operations are atomic, and don't have this problem of everything being liked together. It is easier to maintain and to evolve. Things are as clear as they are separated.

11. [Q11] In a RDBMS What is the difference between a Clustered and a Non-Clustered Index

Indexes are used to speed up searches, you look in the index first, instead of on the actual content.

Clustered indexes only sort tables, there is only one per table, which means they don’t consume extra storage space. They are faster because there is a single lookup step (since they are in the table). They define how data is sorted on a table.

Non-clustered indexes are stored outside the actual table, so they occupy more space. There can be more than one per table, but they involve an extra lookup step (since they are not in the actual table), which makes them slower.

12. [Q12] What are the advantages/disadvantages of user Stored Procedures

It is basically a trade-off between execution performance and memory allocation.

There are other aspects, because stored procedures can be harder to create and maintain, and they are also dependant on the database system. On the other hand, they can improve security because the users with access can be controlled, and also it is not susceptible to injection attacks.

F. Web Development

1. [Q01] Explain how the keyword **‘this’** works in JavaScript. Can you give an example of one of the ways that working with **‘this’** has changed in ES6?

It is a reference to the own object it belongs to, so it depends on the scope.

In the scope of a method, it refers to the instantiated object.

Alone or in a function it refers to the global object.

In a function, in strict mode it is undefined.

In ES6 Arrow Functions **'this'** refers to the originating context of the arrow function, because arrow functions don't bind their own context, so they don’t bind their own ‘**this’**.

2. [Q02] Explain how prototypal inheritance works.

JavaScript is the only popular, widely used language that uses prototypal inheritance, that I can think of, so I am talking about this concept related to this language.

In JavaScript, all objects have a hidden [[Prototype]] property that's either another object or null. Functions are also objects in JavaScript, which means they also have this property. The object being referenced by this property is called the prototype.

If you call a method from an object and it does not exist, the engine will work for that method on the prototype. In this scenario, the 'this' that has been talked about in the previous question, is a reference to the prototype object.

It works by a chain lookup. Instead of defining an abstraction through a class, the programmer simple creates an object, which can be reused by new objects, thanks to this mechanism.

3. [Q03] What's the difference between a variable that is: `null`, `undefined` or undeclared? How would you go about checking for any of these states?

undeclared means a variable has not been declared. Check it by catching an exception for ReferenceError. But I can also be done with the code below, which will catch both if it is undeclared or undefined:

```
if (typeof variableName !== 'undefined')
```

undefined means a variable has been declared but has not yet been assigned a value:

```
if (variable === undefined)
```

null is an assignment value. It can be assigned to a variable as a representation of no value:

```
if (variable === null)
```

the code bellow is what I usually use, i twill check if the variable exists and will catch null and undefined simultaneously :

```
if (variable == null)
```

which is equivalent to:

```
if (variable === undefined || variable === null)
```

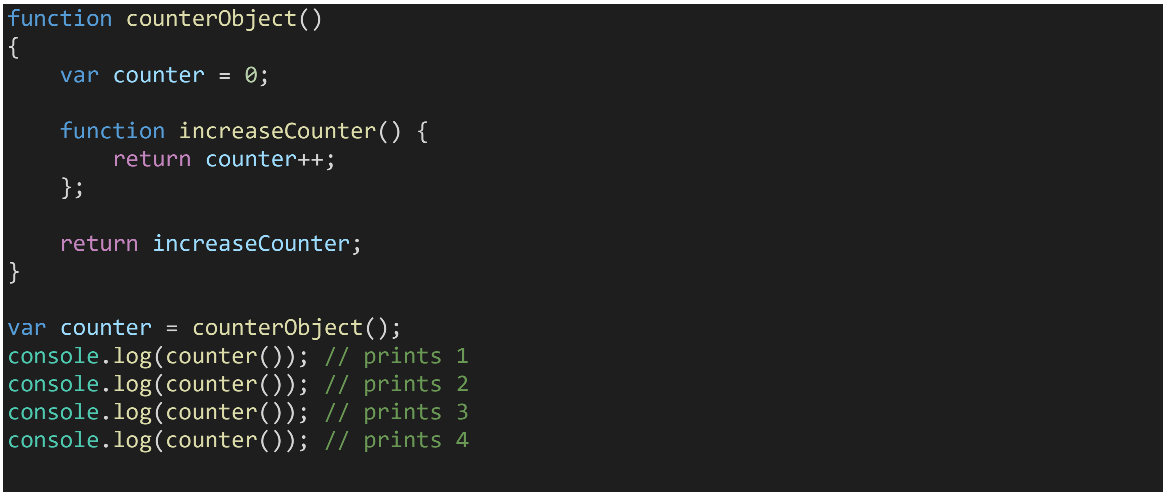
4. [Q04] What is a closure, and how/why would you use one?

In JavaScript functions are objects. You can have functions inside functions, or a function referenced by a variable and passed as an argument to another function.

Because of closure, an inner function always has access to the variables and arguments of its outer function, even after the outer function has returned (or closed).

So it can be seen as a way of making functions access variables that are outside of their scope. Kind of a system of ‘privileged, ‘public’ or ‘private’ variable types. So closure is useful to create private variables or functions.

Basically a function can access variables outside of it. What happens if an outer variable changes, does the function gets the most recent value or the value from when it was instantiated? The code below has the answer :



```
function counterObject()
{
    var counter = 0;

    function increaseCounter() {
        return counter++;
    };

    return increaseCounter;
}

var counter = counterObject();
console.log(counter()); // prints 1
console.log(counter()); // prints 2
console.log(counter()); // prints 3
console.log(counter()); // prints 4
```

5. [Q05] What are the differences between variables created using `let`, `var` or `const`?

They are keywords used in variable declaration. They are different ways of declaring variables.

A variable declared with var exists on the function/object scope.

A variable declared with let exists on the block scope. So it could exist in an if block, but not on the function block, for example. So, it exists in the middle of some {}s.

A variable declared with const is like let because it only exists on the block scope. But you are also not able to change it’s value once it has been assigned, because it is “constant”.

6. [Q06] Explain what is Typescript and how is it different from JavaScript?

It can be seen as JavaScript but better. TypeScript is a superset of JavaScript which primarily provides optional static typing, classes and interfaces. One of the big benefits is to enable IDEs to provide a better environment for detecting small errors as you type the code or, in other words, static analysis of the code, or even complete linting. Being a superset of JavaScript, means that any JavaScript code can be used in TypeScript. It is still deployable as a JavaScript application. Ah, and it’s open source.

7. [Q07] Can an interface extends a class just like a class implements an interface?

In TypeScript, an interface can extend a class, because the compiler takes care of figuring out the type signature for the class and it also adds the signature to the interface. When the interface is implemented, the methods and properties explicitly declared in the interface, and the methods and properties extracted from the original class, need to be implemented.

I usually tell people that the answer is always yes to questions like ‘can X be done in technology?’. It might be just a matter of time and effort to make it possible!

8. [Q08] Explain some of the benefits of using Typescript

It can take more effort than writing JavaScript code, but in the end it can pay off, because the development environment can help the developer more to avoid errors that would only be detected on execution time. The biggest benefit is really that: better tool support to aid the development, which leads to more robust software, specially in big projects.

The code can be automatically documented, bugs can be caught in advanced. And there is virtually no learning curve, if someone already knows JavaScript.

9. [Q09] Explain according to your experience what are the biggest strengths and weaknesses of Javascript

It is the programing language with more lines of code executed per day in the planet, but it can have a weird syntax, sometimes.

It is very powerful, yet friendly to new programmers. For example, it is object oriented, but people can do all sorts of things with it without understanding polymorphism or even what is a class.

For local development it is kind of tricky to install an engine to run code locally, and there are many engines available, with different compatibilities and features, this can cause environment problems on deploys.

It can be used to execute code on the client’s machine, which can be great but also dangerous for everyone. The client could execute malicious code, such as a crypto currency miner, and the server can’t rely on the data the client outputs, because the code can be easily modified by the client.

H. Practical Questions

For the following section you can use the language of your choice, examples are written in Java, but they can be changed according to your preferences.

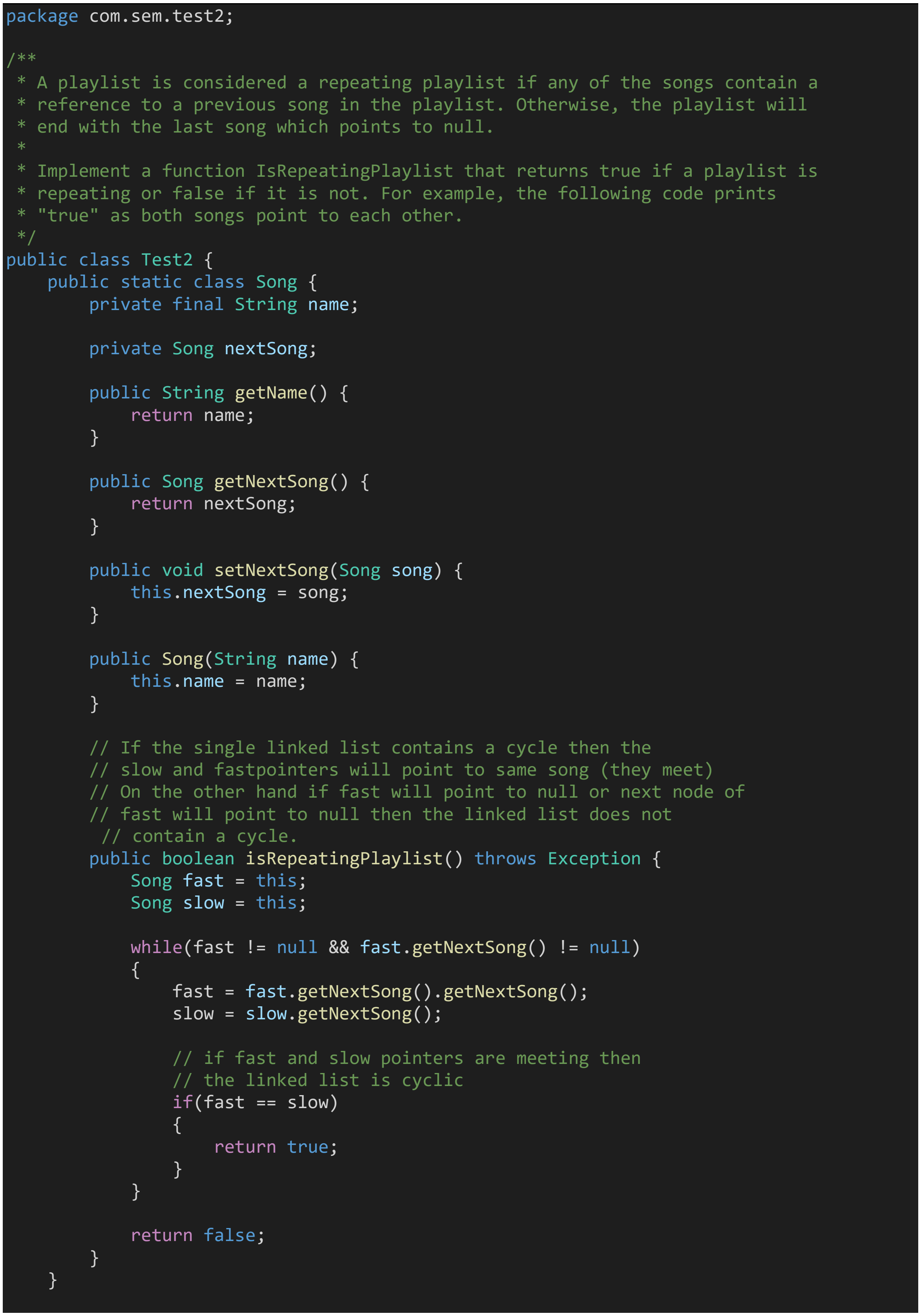
1. [Q01] Implement the Union Method in the following case

```
package com.sem.test1;

import java.util.Arrays;
import java.util.HashSet;
import java.util.stream.Stream;

/**
 * Implement the union method. When passed two arrays of names, it will
 * return an array containing the names that appear in either or both arrays.
 *
 * The returned array should have no duplicates.
 *
 * For example, calling
 *
 * union(new string[]{'Ava', 'Emma', 'Olivia'}, new string[]{'Olivia',
 * 'Sophia', 'Emma'})
 *
 * should return an array containing Ava, Emma, Olivia, and Sophia in any order.
 */

public class Test1 {

    public static void main(String[] args) throws Exception {
        String[] names1 = new String[] { "Ava", "Emma", "Olivia" };
        String[] names2 = new String[] { "Olivia", "Sophia", "Emma" };

        System.out.println(Arrays.toString(union(names1, names2)));
    }

    public static String[] union(String[] names1, String[] names2) throws Exception {
        // I will just concatenate the arrays and then get the set of them,
        // which will take care of duplicates
        String[] concatenated_array = Stream.concat(Arrays.stream(names1),
Arrays.stream(names2)).toArray(String[]::new);
        String[] names_array = new
HashSet<String>(Arrays.asList(concatenated_array)).toArray(new String[0]);

        return names_array;
    }
}
```

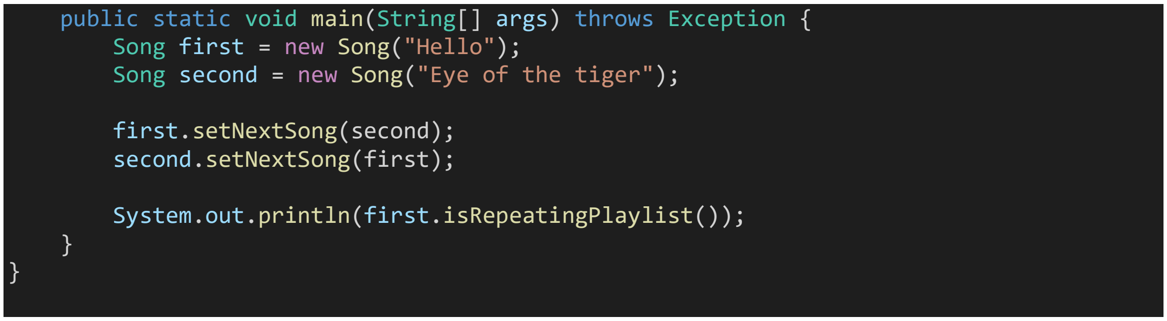
My preferred language is python, so I also implemented in it:

```
def union(names1, names2):
    # concatenate the arrays
    concatenation = names1 + names2

    # get the set of the concatenated array and return
    return set(concatenation)


if __name__ == "__main__":
    names1 = ["Ava", "Emma", "Olivia"]
    names2 = ["Olivia", "Sophia", "Emma"]

    print(union(names1, names2))
```

2. [Q02] Implement the specified function

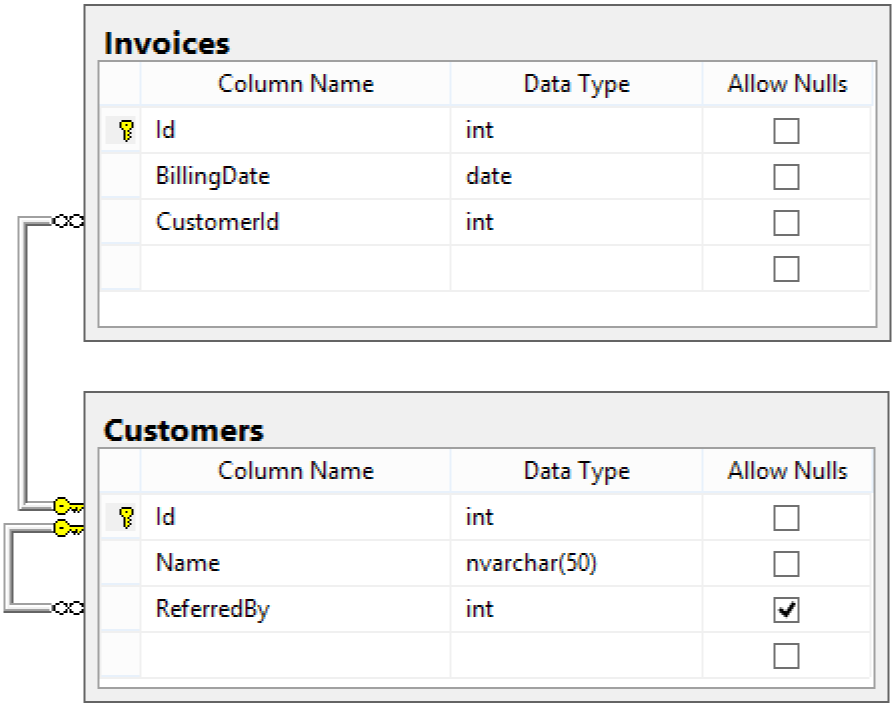


```
package com.sem.test2;

/**
 * A playlist is considered a repeating playlist if any of the songs contain a
 * reference to a previous song in the playlist. Otherwise, the playlist will
 * end with the last song which points to null.
 *
 * Implement a function IsRepeatingPlaylist that returns true if a playlist is
 * repeating or false if it is not. For example, the following code prints
 * "true" as both songs point to each other.
 */
public class Test2 {
    public static class Song {
        private final String name;

        private Song nextSong;

        public String getName() {
            return name;
        }

        public Song getNextSong() {
            return nextSong;
        }

        public void setNextSong(Song song) {
            this.nextSong = song;
        }

        public Song(String name) {
            this.name = name;
        }

        // If the single linked list contains a cycle then the
        // slow and fastpointers will point to same song (they meet)
        // On the other hand if fast will point to null or next node of
        // fast will point to null then the linked list does not
         // contain a cycle.
        public boolean isRepeatingPlaylist() throws Exception {
            Song fast = this;
            Song slow = this;

            while(fast != null && fast.getNextSong() != null)
            {
                fast = fast.getNextSong().getNextSong();
                slow = slow.getNextSong();

                // if fast and slow pointers are meeting then
                // the linked list is cyclic
                if(fast == slow)
                {
                    return true;
                }
            }

            return false;
        }
    }
```



```
    public static void main(String[] args) throws Exception {
        Song first = new Song("Hello");
        Song second = new Song("Eye of the tiger");

        first.setNextSong(second);
        second.setNextSong(first);

        System.out.println(first.isRepeatingPlaylist());
    }
}
```

3. [Q03] Considering the database schema displayed in the SQLServer-style diagram below, write a SQL query to return a list of all the invoices. For each invoice, show the Invoice ID, the billing date, the customer's name, and the name of the customer who referred that customer (if any). The list should be ordered by billing date.

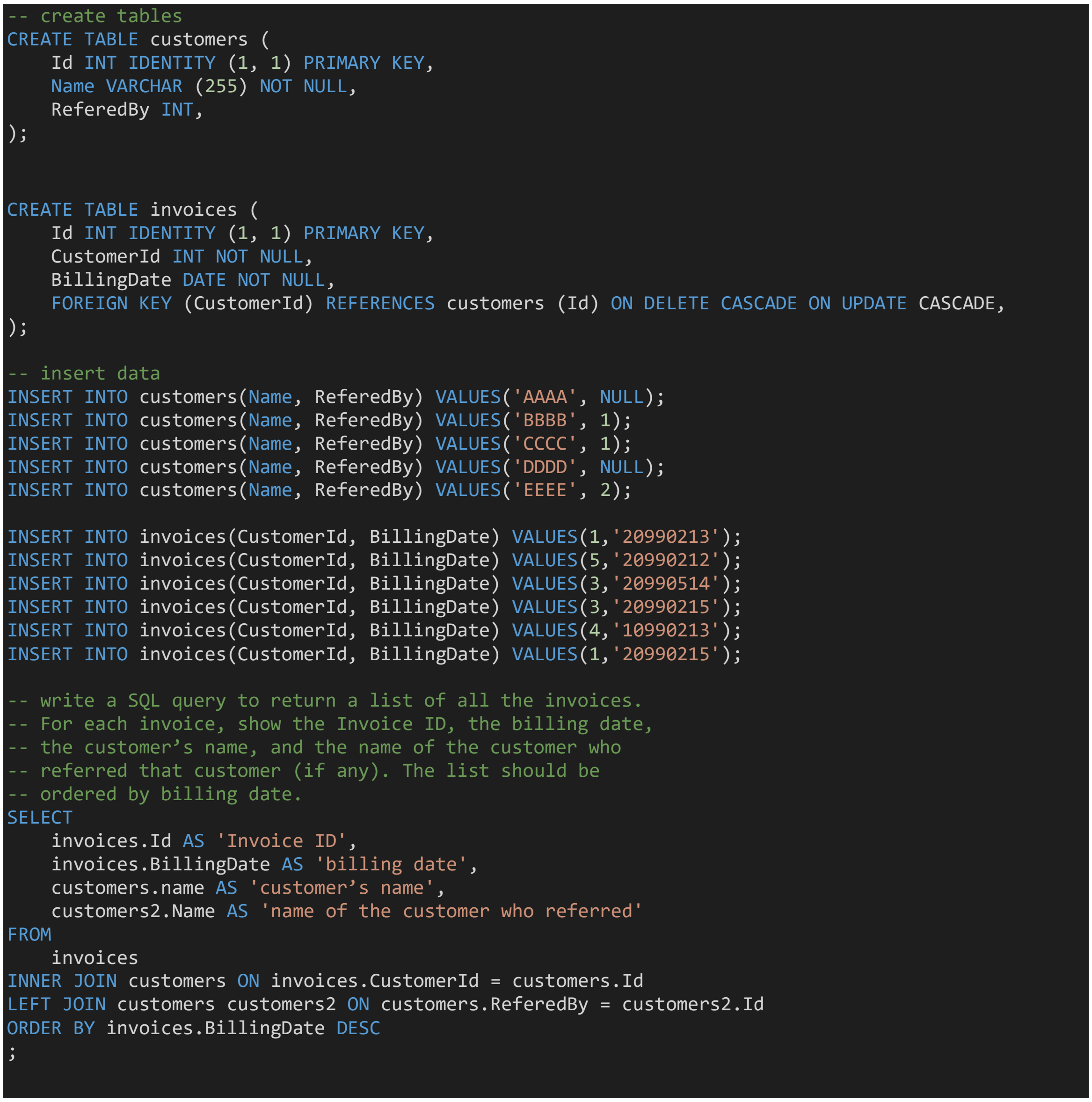


Invoices

	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	BillingDate	date	☐
	CustomerId	int	☐
			☐

Customers

	Column Name	Data Type	Allow Nulls
🔑	Id	int	☐
	Name	nvarchar(50)	☐
	ReferredBy	int	☑
			☐



```sql
-- create tables
CREATE TABLE customers (
    Id INT IDENTITY (1, 1) PRIMARY KEY,
    Name VARCHAR (255) NOT NULL,
    ReferedBy INT,
);


CREATE TABLE invoices (
    Id INT IDENTITY (1, 1) PRIMARY KEY,
    CustomerId INT NOT NULL,
    BillingDate DATE NOT NULL,
    FOREIGN KEY (CustomerId) REFERENCES customers (Id) ON DELETE CASCADE ON UPDATE CASCADE,
);

-- insert data
INSERT INTO customers(Name, ReferedBy) VALUES('AAAA', NULL);
INSERT INTO customers(Name, ReferedBy) VALUES('BBBB', 1);
INSERT INTO customers(Name, ReferedBy) VALUES('CCCC', 1);
INSERT INTO customers(Name, ReferedBy) VALUES('DDDD', NULL);
INSERT INTO customers(Name, ReferedBy) VALUES('EEEE', 2);

INSERT INTO invoices(CustomerId, BillingDate) VALUES(1,'20990213');
INSERT INTO invoices(CustomerId, BillingDate) VALUES(5,'20990212');
INSERT INTO invoices(CustomerId, BillingDate) VALUES(3,'20990514');
INSERT INTO invoices(CustomerId, BillingDate) VALUES(3,'20990215');
INSERT INTO invoices(CustomerId, BillingDate) VALUES(4,'10990213');
INSERT INTO invoices(CustomerId, BillingDate) VALUES(1,'20990215');

-- write a SQL query to return a list of all the invoices.
-- For each invoice, show the Invoice ID, the billing date,
-- the customer’s name, and the name of the customer who
-- referred that customer (if any). The list should be
-- ordered by billing date.
SELECT
    invoices.Id AS 'Invoice ID',
    invoices.BillingDate AS 'billing date',
    customers.name AS 'customer’s name',
    customers2.Name AS 'name of the customer who referred'
FROM
    invoices
INNER JOIN customers ON invoices.CustomerId = customers.Id
LEFT JOIN customers customers2 ON customers.ReferedBy = customers2.Id
ORDER BY invoices.BillingDate DESC
;
```

Please write the time you spend in hours in this exam:

I didn’t count them, I divided the time between 3 days, and in total it must have been around 15 hours! I took it slow, allowing me to learn more about some concepts I haven’t been used to, develop skills with technologies I haven’t used in a while and also to double check my answers.

X. MISCELLANEOUS QUESTIONS

[Q1] How would you change this exam and/or examination process?

OK, so the following is all my own opinions!

First of all, I actually had fun doing this exam. Felt like the old university days =)

Many questions were asked to see if I knew something or not. In other words, if I remember something. I just googled for all of the answers to make sure they were right. Memory for humans is a dynamic thing, and should not be trusted!

So, if I am dealing with a problem and I have the possibility of double checking what I think I know, I will do it. Everyone should do it, after all, there is no absolute truth in science, which means things can change, specially if we're talking about the state of the art.

With this in mind, in my opinion, some questions are not really useful for testing valuable skills of your candidates, because anyone can search for an answer online or even worse, anyone can just remember a text they read. In the past, knowing or remembering was a valuable skill, after all knowledge was hard to get: you had to find the book, or find the person that knew it. Nowadays there is an abundance of resources, and the skilled person is that person that is capable of doing good searches and also filter well the results!

Furthermore, this is the computer science field, not law, for example. By this I mean that a 60 years old lawyer can be seen as a very experienced person that knows everything and so is a very valuable asset to a team, while a 60 years old developer not necessarily knows everything, because the technologies in the field are in constant change, and things become obsolete every day. So, while a good lawyer has good memory, a good computer scientist has the ability to learn, and not become stagnated with what they already knew from past studies or experience.

When I was the one conducting interviews to hire people for development roles (or any role for that matter), I wanted to basically test them for three things:

1. how they search for knowledge?
2. how fast they can learn something new?
3. how can they solve problems?

Which means I literally was interested in how they use search engines like DuckDuckGo (or even the not so recommended Google). I would also try to present them with bizarre concepts that you don't always find in computer science courses or others alike. Finally, the test questions would be centered around raw problem solution, independent of technology, after all the technologies change with time, so better than knowing something that can become obsolete at any time, I am more interested in people that can learn things, because those are the ones that are capable of staying on top of the game.

One more thing, is that it could be nice to implement the test in a platform that auto-corrects the test, allows the candidate to choose whichever language he wants for the implementation questions and also collects data from the candidate, like how long it took him to complete it. I can set up a web server on it to achieve this very same thing in 4 hours (click here to see what I am talking about), but there are plenty available, like hackerrank.com or leetcode.com.