Question a:

Q: In your own words, explain what is meant by the functional paradigm in computer programming? What are the perceived advantages of this programming paradigm?

A:

Functional programming means computation by pure function calls that avoid any global mutable state or side effects. The big idea in functional programming is the separation of the functions that operate on the data from the data on which they operate. This separation lends itself to the idea of function composition. Functions operate on data, transforming each stage and putting the results on to the next phase.

The first benefit is that the resultant composition can be implemented and can implement potentially complex transformations in a very readable and maintainable way. And the functional expression of a problem is often more obvious and easier to reason about.

Question b:

Q: List five features of the JavaScript language which allows it to support functional programming.

A:

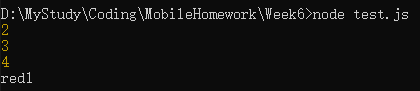
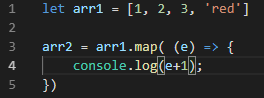
1. JavaScript support higher order functions.
2. JavaScript support function closures.
3. JavaScript support curried functions.
4. JavaScript support lazy evaluation and immutability through the const keyword.
5. Though JavaScript does not support the native structural sharing, developers can realize it using third party libraries.

Question c:

Q: In functional programming, what is meant by the term functor? In your answer show, using JavaScript code, an example of a functor.

A:

A functor is a data object that can hold elements of any data type which implements the map operation or function.



As is seen in the picture, arr1 is an array with 3 numbers and a string inside, we can use map operation where for each element of the array, output the element + 1. For the 3 numbers, output another number which is the original number plus 1, while for the string, output the string following a character ‘1’. So arr is a so-called functor in JavaScript. Because arr hold two kinds of elements and can implement map operation.

Question d:

Q: What is the visitor pattern? Describe how the visitor pattern is implemented on JavaScript arrays.

A:

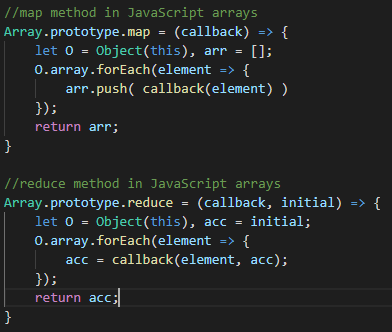
Visitor pattern is an operation operating on each element of a specific data structure, or the so-called functor. But as functors are immutable, so the results of visitor patterns will come to a new container with the same data structure as the functor.

The array for each method implements the visitor pattern calling a specified function for each element of JavaScript arrays. Now arrays come with a for each method already defined. For each implements a general visitor pattern that can be built on.

Question e:

Q: Explain the difference between the map operation and the reduce operation, illustrating your answer with JavaScript code.

A:



And as is seen in the picture, there are some difference between map method and reduce method.

First, they have different numbers of argument. Map method only takes a callback function as an argument, while reduce method takes a callback function and an initial value as at least two arguments.

Second, they do not return the same type of value. The map method must return an array as a result. On the other hand, the return value of reduce method is up to the data type of the initial seed value. So it can return a string, an array or a number as well.