1 Extending Array Prototype (40 marks)

Implement the following recursive methods to extend JavaScript Array prototype:

- **cons**: This method takes one argument of any value and returns an array with the value inserted in front of the original array.

- **foldr**: This method takes two arguments, a dyadic function and a default value. When evaluated, foldr places the dyadic function between each element of the original array and adds the default at the end. It then returns the result of evaluating the expression from right to left.

- **foldl**: This method takes two arguments, a dyadic function and a default value. When evaluated, foldl places the dyadic function between each element of the original array and adds the default at the front. It then returns the result of evaluating the expression from left to right.

2 Regular Expressions (60 marks)

Implement a JavaScript program that takes the following text inputs from a user through web interface such as FORM. Implement a function to validate the text inputs when a "submit" button is clicked. Display the information that are valid and report those that are invalid.

2.1 Name on Credit Card

This includes last and first name. Each of the last and first names is at least 2 alphabets long and starts with a capital letter. They are separated by ,.
2.2 Credit Card Number

For simplicity, a credit card number is only 8-digit instated of 16-digit long. The last digit of a credit card number is the check digit, which protects against transcription errors such as an error in a single digit or switching two digits. The following method is used to verify a 8-digit credit card numbers:

- Starting from the rightmost digit, form the sum of every other digit. For example, if the credit card number is 4358 9795, then you form the sum $5 + 7 + 8 + 3 = 23$.

- Double each of the digits that were not included in the preceding step. Add all digits of the resulting numbers. For example, with the number given above, doubling the digits, starting with the next-to-last one, yields 18 18 10 8. Adding all digits in these values yields $1 + 8 + 1 + 8 + 1 + 0 + 8 = 27$.

- Add the sums of the two preceding steps. If the last digit of the result is 0, the number is valid. In our case, $23 + 27 = 50$, so the number is valid.

2.3 Credit Card Expiration Date

The expiration date is in MM/DD/YYYY format, where M, D and Y are digits.

2.4 E-mail address

The e-mail address starts with alpha-numeric characters (both lowercase and uppercase characters are allowed). It may have periods, underscores and hyphens. This is followed by @ symbol, then 2 to 8 alpha-numeric characters, where the alphabets can be either lowercase or uppercase. After that, there should be a period . symbol, then a domain name that has 2 to 4 alphabets.