

Haskell Programming Exercises

1. Write a function *insertall* that inserts a specified element into all possible positions in a list. For example, *insertall* would insert a 1 into the list [2,3,4] to obtain

[[1,2,3,4],[2,1,3,4],[2,3,1,4],[2,3,4,1]]

2. Matrix multiplication of two matrices A and B is only possible if the number of rows in A is the same as the number of columns in B. For example,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} (ae + bg) & (af + bh) \\ (ce + dg) & (cf + dh) \end{bmatrix}$$

Write a function that multiplies any two valid matrices.

3. Write a function *riffle* that takes a list as a single argument. It is assumed that the list contains an equal number of even and odd integers. The function produces a rearrangement of the elements in the list so that odd and even number alternate. If the list contains a different number of even and odd integers, the list returned will contain twice as many integers as there are odd numbers, if there are less odd numbers or twice as many integers as there are even numbers, if there are less even numbers. For example,

riffle [2,4,5,3,1,6] → [2,5,4,3,6,1]
riffle [2,4,5,3,1,6,7] → [2,5,4,3,6,1]
riffle [2,4,5,3,1,6,8] → [2,5,4,3,6,1]

4. The function **minimum** extracts the minimum value from a list. Using *minimum*, ++ and list comprehensions, we can take a list of numbers and return a list of numbers which is sorted from lowest to highest. Write the function *sortit* that will take an unsorted list and return a sorted list using the conditions above.