The following problems should be solved in Miranda. In all cases, provide the **signature** of your function along with its definition. Submit your solutions and any script files that demonstrate correctness by using the `turnin` program. Recall that comments in Miranda can be specified using double bars (||) and that Miranda resides in `/p/bin/mira`.

1. Write a function `MinMax` that takes a list of numbers as an argument and returns a tuple in which the first element is the smallest number of the list and the second element is the largest element of the list.

2. (a) Using recursion, write a function `double` that duplicates each element of a list. For example, `double "abc"` should return `"aabbcc"`.

(b) Write the same function using a list comprehension.

(c) Write the same function using `foldr`.

3. (a) Use recursion to write a function called `mysubtract` that takes two lists as arguments and behaves the same as the list subtraction operator (`--`) except that elements are subtracted from in right to left order. Examples of correct behavior are shown below:

```
Miranda [1, 2, 1] -- [1]
[2, 1]

Miranda mysubtract [1, 2, 1] [1]
[1, 2]

Miranda mysubtract [1, 2] [2, 1]
[]
```

(b) Use `foldr` or `foldl` to write the `mysubtract` function.

4. A perfect number is one in which the sum of its factors that are less than itself equals the number itself. For example, the factors of 28 are $1, 2, 4, 7$ and 14. Since $1 + 2 + 4 + 7 + 14 = 28$, 28 is a perfect number. By contrast, the factors of 8 are $1, 2$ and 4, so 8 is not a perfect number. Write a `perfect` function, which takes a number as a parameter and returns a list of all perfect numbers between 1 and that number.

5. Without using explicit recursion, define a function called `increasing` which takes a list of numbers and returns `True` if the list of numbers is sorted in increasing order. Examples of the function's behavior are given below:

```
Miranda increasing [3, 2, 1]
False

Miranda increasing [1, 2, 2, 4]
False

Miranda increasing [3, 4, 5]
True
```