Exercises
List Comprehensions

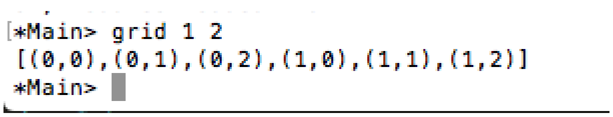
1. Using a list comprehension, give an expression that calculates the sum of

$$\sum_{i=1}^{\imath=100} i^2$$

2. Suppose that a *coordinate grid* of size $m \times n$ is given by the list of all pairs (x, y) of integers such that $0 \leq x \leq m$ and $0 \leq y \leq n$. Using a list comprehension, define a function:

```
grid :: Int -> Int -> [(Int, Int)]
```

that returns a coordinate grid of a given size. For example:

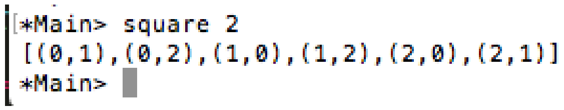


```
*Main> grid 1 2
[(0,0),(0,1),(0,2),(1,0),(1,1),(1,2)]
*Main>
```

3. Using a list comprehension and the function **grid** above, define a function

```
square :: Int ->  [(Int, Int)]
```

that returns a coordinate square of size n, excluding the diagonal from $(0, 0)$ to (n, n). For example:

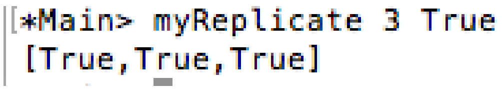


```
*Main> square 2
[(0,1),(0,2),(1,0),(1,2),(2,0),(2,1)]
*Main>
```

4. In a similar way to the function ***length***, show how the library function

```
replicate :: Int -> a -> [a]
```

that produces a list of identical elements can be defined using list comprehension. (Call your version **myReplicate**) For example:

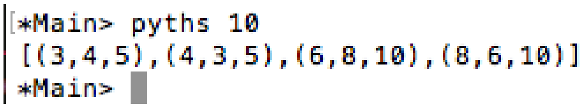


```
*Main> myReplicate 3 True
[True,True,True]
```

5. A triple (x, y, z) of positive integers is called pythagorean if $x^2 + y^2 = z^2$. Using a list comprehension, define a function

```
pyths :: Int -> [(Int,Int,Int)]
```

that returns a list of all such triples whose components are at most a given limit. For example

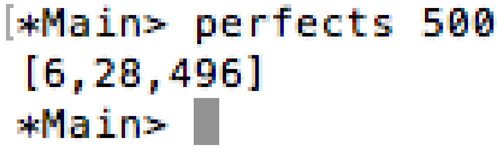


```
*Main> pyths 10
[(3,4,5),(4,3,5),(6,8,10),(8,6,10)]
*Main>
```

6. A positive integer is perfect if it equals the sum of all of its factors, excluding the number itself. Using a list comprehension and the function **factors**, define a function

```
perfects :: Int -> [Int]
```

that returns the list of all perfect numbers up to a given limit. For example:



```
*Main> perfects 500
[6,28,496]
*Main>
```