1. This week's Problem of the Week in Math is described as follows:

> *There are thirty positive integers less than 100 that share a certain property. Your friend, Blake, wrote them down in the table to the left. But Blake made a mistake! One of the numbers listed is wrong and should be replaced with another. Which number is incorrect, what should it be replaced with, and why?*

The numbers are listed below.

6	10	14	15	21
22	26	33	34	35
38	39	46	51	55
57	58	62	65	69
75	77	82	85	86
87	91	93	94	95

Use the fact that the "certain" property is that these numbers are all supposed to be the product of *unique* prime numbers to find and fix the mistake that Blake made.

Reminder: Code your solution in an R script and copy it over to this `.Rnw` file.

Hint: You may find the `%in%` operator and the `setdiff()` function to be helpful.

Solution:

```r
#################
## Making set of prime numbers
#################
prime = c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53,
          59, 61, 67, 71, 73, 79, 83, 89, 97) #Set of all prime numbers under 100

wrong.vec = c(6, 10, 14, 15, 21, 22, 26, 33, 34, 35, 38, 39, 46, 51, 55, 57, 58,
              62, 65, 69, 75, 77, 82, 85, 86, 87, 91, 93, 94, 95)

prime.mult = c() #setting up what will be final vector
prime.index = c(1:length(prime))

####################################
##  Making a vector of unique prime multiples
####################################

for (i in prime.index){ #Loop over each prime to add itself and unique multiples to vector
  curr = prime[i]
  #Initializing multiples
  count = 1
  mult = curr*prime[i+count] #Initializing first multiple
  while(i<length(prime) & mult<100){ #Adding unique multiples
    if (!mult %in% prime.mult) #Checks if it is unique
      prime.mult = c(prime.mult, mult) #Add to vector if it is unique
    #Prepping for next iteration
    count = count+1
    mult = curr*prime[i+count]
  }
}
###Making the vector sorted properly
prime.mult = sort(prime.mult)

### Comparing given vs calculated vector
setdiff(wrong.vec, prime.mult) #75 should NOT be in given vector
```

```
## [1] 75
```

```r
setdiff(prime.mult, wrong.vec) #74 SHOULD be in given vector
```

```
## [1] 74
```

```r
#From here, replacing either 74 with 75 in given vector
#or replacing the given vector with prime.mult would successfully correct the mistake
```