**Function Programming**

**Oghenenohwo Ojakovo**

**Exercise 1**

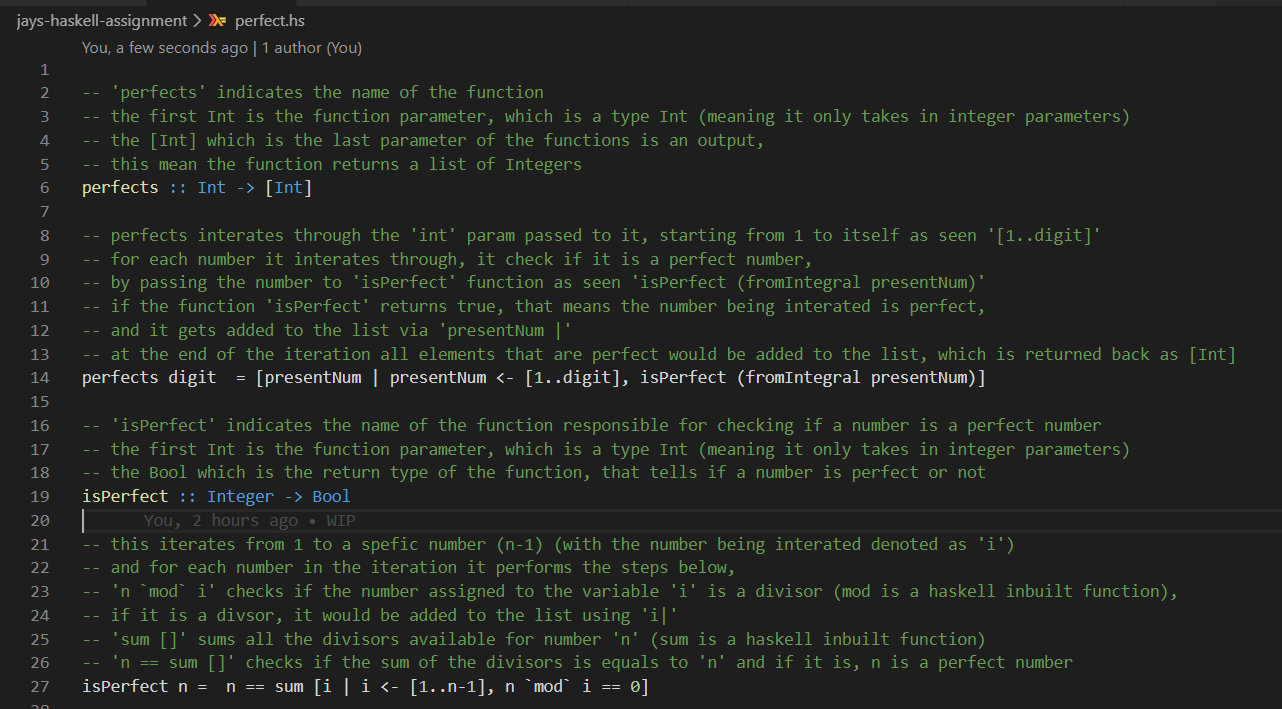
A positive integer is perfect if it equals the sum of all of its factors, excluding the number itself. Using a list comprehension, define a function

perfects:: Int -> [Int]

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

Ghci> perfects 500

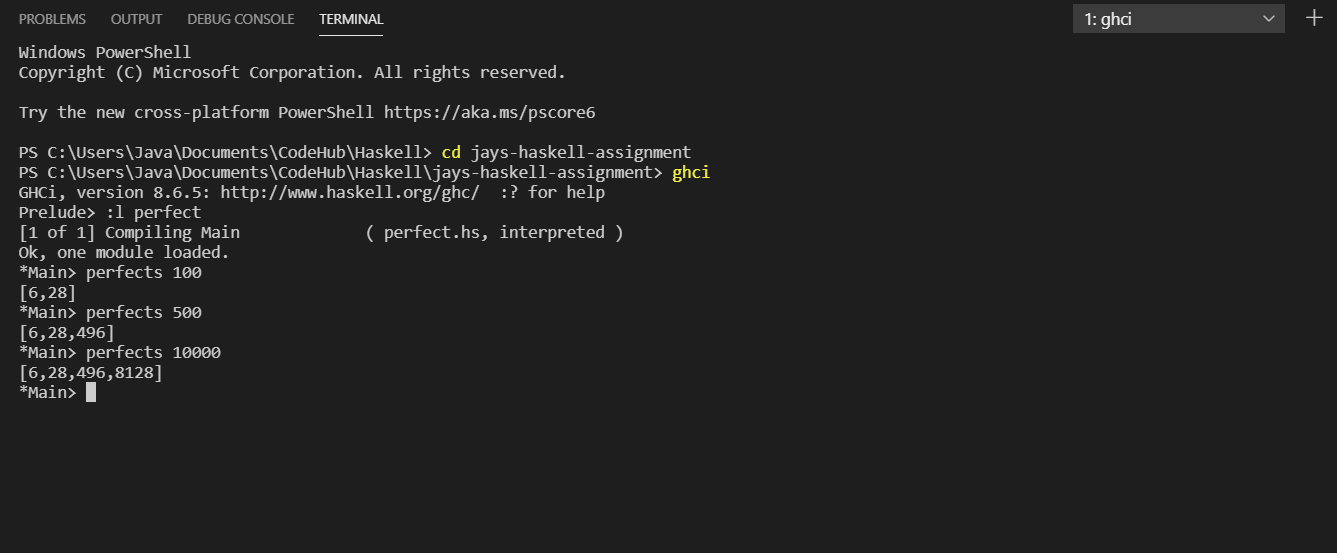
[6,28,496]



An integer is greater than one is a perfect number which will be the positive numbers which are one and the number itself. Hence, using the perfects, a simple function that will decides if an integer is perfect which can be defined.

Here is the output on WinGHCi,

Once I run the code then to call the function as perfects and the number will be put as 500 and as a result, it will show the next number which is [6,28,496]. However, as I entered another number like 10,000, it will show the next number which is perfect. Hence, the code works perfectly.



**Exercise 2**

Write a Caesar Cipher program: The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials. Thus, to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down. The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

Examples :

Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ

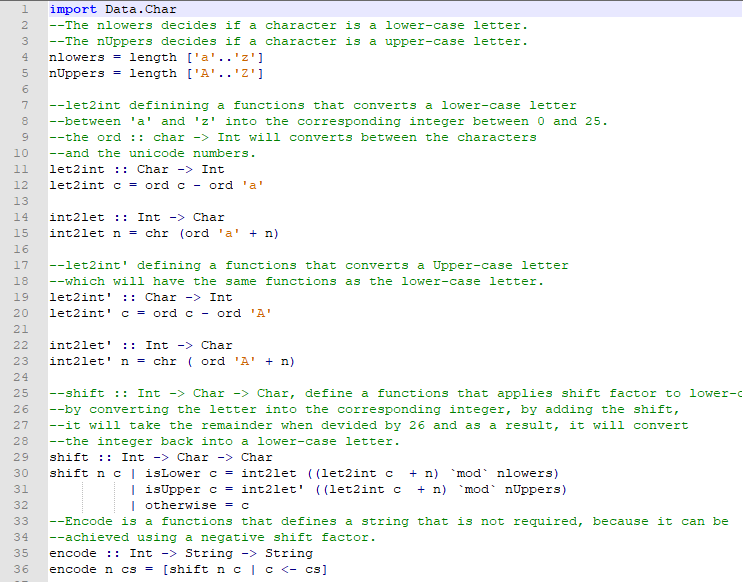
Shift: 23

Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE

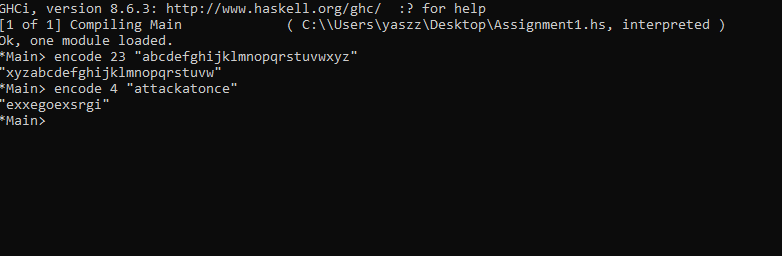
Shift: 4

Cipher: EXXEGOEXSRGI



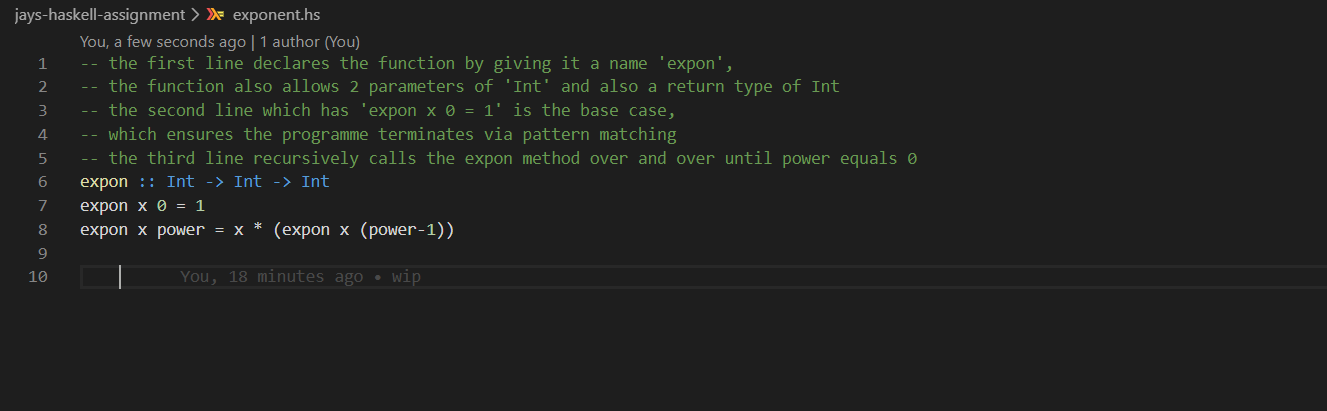
Implementing a Caesar cipher,is to encode the string which simply replaced each letter in the string by the letter that will places further down and wrapping aorund the alphabet until the end of the alphabet. However, the use Caesar cipher can easily cracked by exploiting information about letter frequencies in english text.

As you can see the output below in WinGHCi, the alphabet will encrypt evey letter in the message and each letter of the plain text message will be replaced.

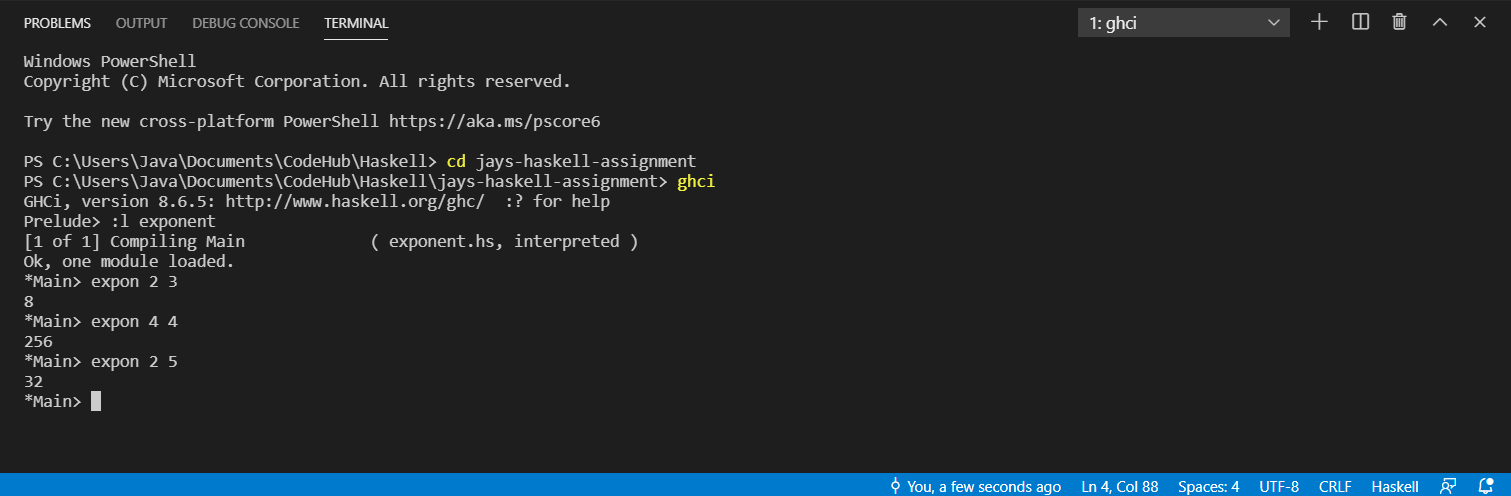


**Exercise 3**

Define the exponentiation operator ^ for non-negative integers using the same pattern of recursion as the multiplication operator \*, and show how 2 ^ 3 is evaluated using your definition.



here as you can see, it works properly as I entered 4^4, 2^5 and it shows the result.



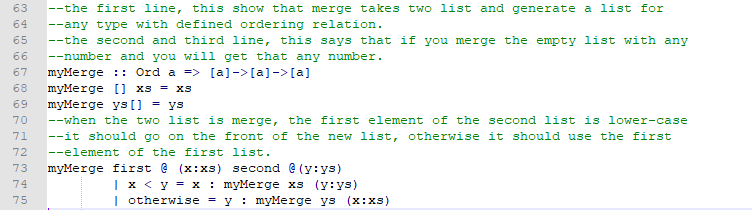
**Exercise 4**

Define a recursive function merge :: Ord a => [a] -> [a] -> [a] that merges two sorted lists to give a single sorted list.

For example: > merge [2,5,6] [1,3,4]

[1,2,3,4,5,6]

Note: your definition should not use other functions on sorted lists such as insert or isort, but should be defined using explicit recursion.



As you can see the output in WinGHCi,I manage the merge [2,3,4][5,6] and as a result, it will merge the number thus if the number is lower then it will be the first number on first list.

