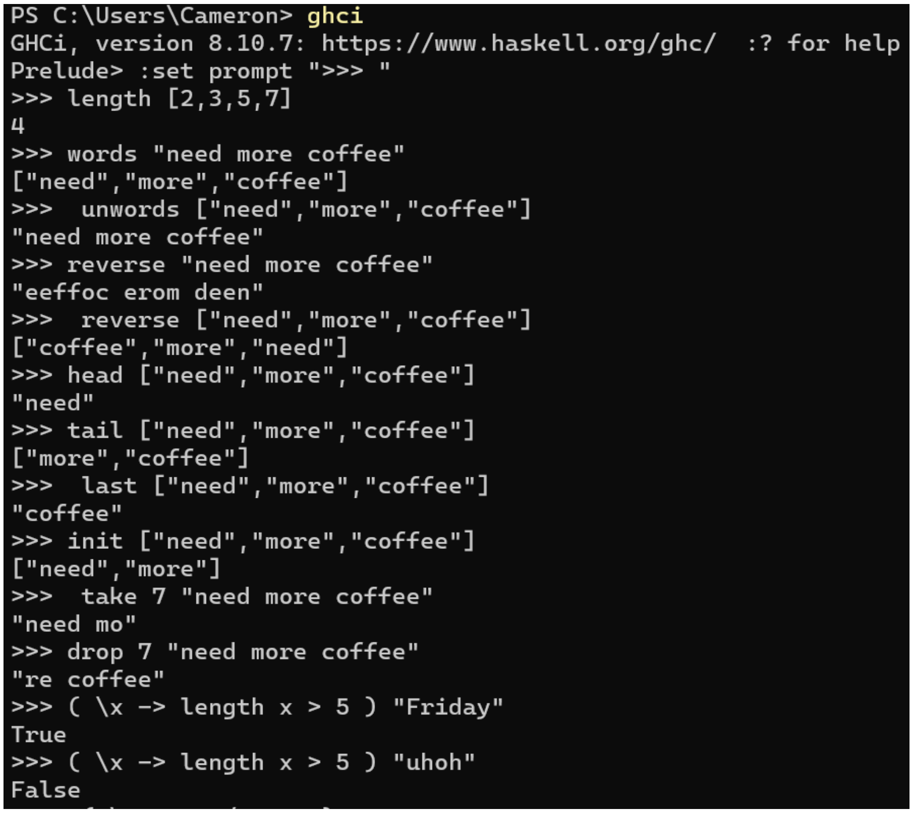
Haskell Programming Assignment: Various Computions:

Cameron Francois
May 5th, 2023
CSC 344

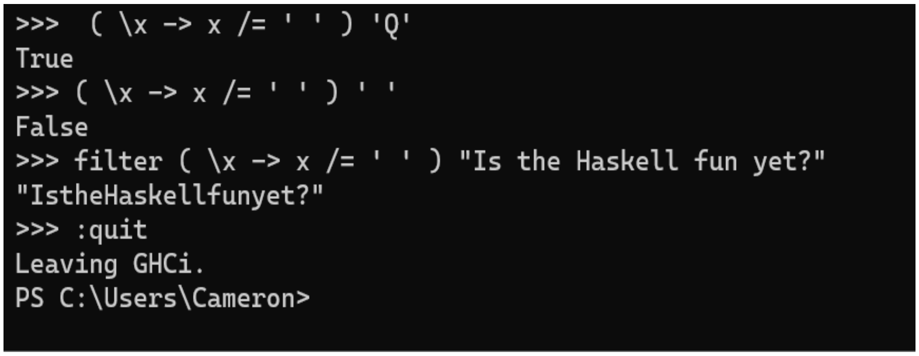
Abstract:

This assignment has us completing 8 tasks in the programming language Haskell. Task 1 requires us to mimic some demos to get a better understanding of how it works. Task 2 and 3 has us writing code on various functions then demo. Task 4 focused on recursive functions and Task 5 dealt with list comprehension. Task 6 had us using higher order functions and task 7 made us use nPVI formula which was interesting to see how it functions. Task 8 had us focusing on demo written code to get a full grasp how Haskell programming functions

Task 1: Mindfully Mimicking the Demo:



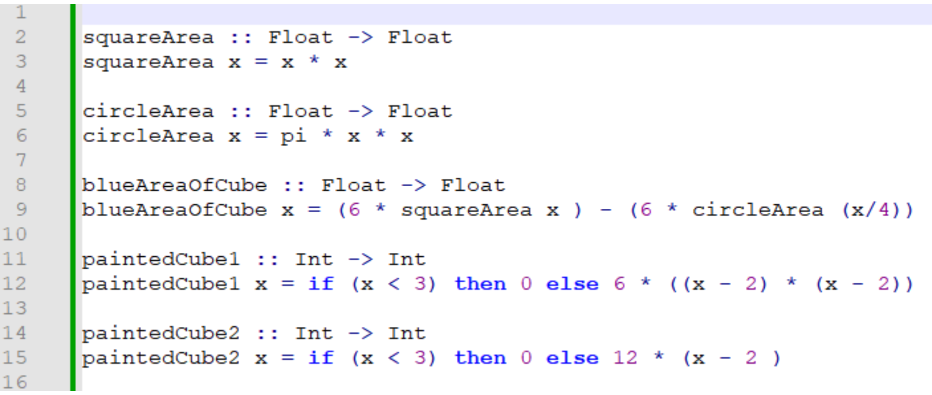
```
PS C:\Users\Cameron> ghci
GHCi, version 8.10.7: https://www.haskell.org/ghc/  :? for help
Prelude> :set prompt ">>> "
>>> length [2,3,5,7]
4
>>> words "need more coffee"
["need","more","coffee"]
>>>  unwords ["need","more","coffee"]
"need more coffee"
>>> reverse "need more coffee"
"eeffoc erom deen"
>>>  reverse ["need","more","coffee"]
["coffee","more","need"]
>>> head ["need","more","coffee"]
"need"
>>> tail ["need","more","coffee"]
["more","coffee"]
>>>  last ["need","more","coffee"]
"coffee"
>>> init ["need","more","coffee"]
["need","more"]
>>>  take 7 "need more coffee"
"need mo"
>>> drop 7 "need more coffee"
"re coffee"
>>> ( \x -> length x > 5 ) "Friday"
True
>>> ( \x -> length x > 5 ) "uhoh"
False
```



```
>>>  ( \x -> x /= ' ' ) 'Q'
True
>>> ( \x -> x /= ' ' ) ' '
False
>>> filter ( \x -> x /= ' ' ) "Is the Haskell fun yet?"
"IstheHaskellfunyet?"
>>> :quit
Leaving GHCi.
PS C:\Users\Cameron>
```

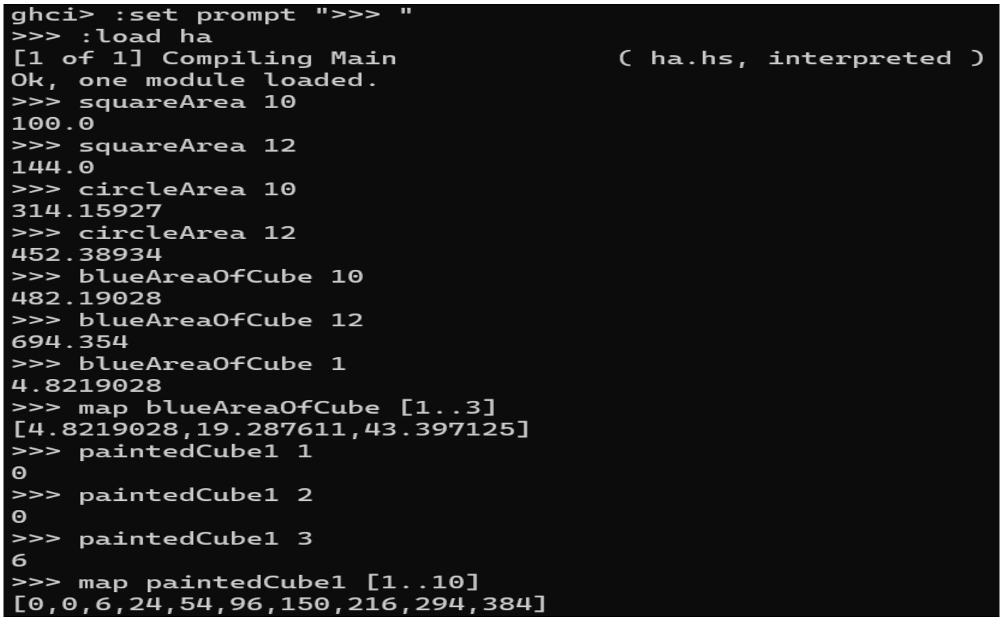
Task 2: Numeric Function Definitions:

Code:

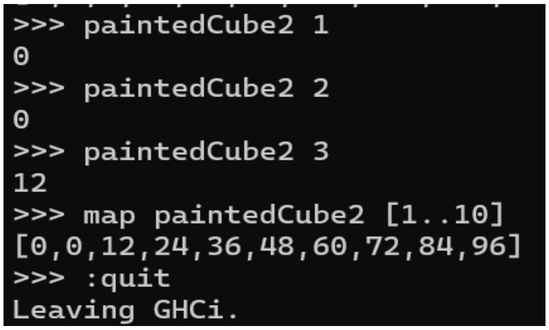


```haskell

squareArea :: Float -> Float
squareArea x = x * x

circleArea :: Float -> Float
circleArea x = pi * x * x

blueAreaOfCube :: Float -> Float
blueAreaOfCube x = (6 * squareArea x ) - (6 * circleArea (x/4))

paintedCube1 :: Int -> Int
paintedCube1 x = if (x < 3) then 0 else 6 * ((x - 2) * (x - 2))

paintedCube2 :: Int -> Int
paintedCube2 x = if (x < 3) then 0 else 12 * (x - 2 )

```

Demo:



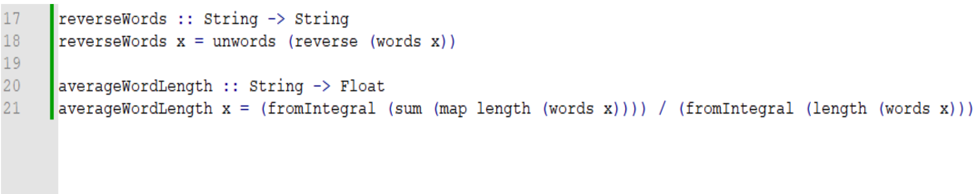
```
ghci> :set prompt ">>> "
>>> :load ha
[1 of 1] Compiling Main             ( ha.hs, interpreted )
Ok, one module loaded.
>>> squareArea 10
100.0
>>> squareArea 12
144.0
>>> circleArea 10
314.15927
>>> circleArea 12
452.38934
>>> blueAreaOfCube 10
482.19028
>>> blueAreaOfCube 12
694.354
>>> blueAreaOfCube 1
4.8219028
>>> map blueAreaOfCube [1..3]
[4.8219028,19.287611,43.397125]
>>> paintedCube1 1
0
>>> paintedCube1 2
0
>>> paintedCube1 3
6
>>> map paintedCube1 [1..10]
[0,0,6,24,54,96,150,216,294,384]
```



```
>>> paintedCube2 1
0
>>> paintedCube2 2
0
>>> paintedCube2 3
12
>>> map paintedCube2 [1..10]
[0,0,12,24,36,48,60,72,84,96]
>>> :quit
Leaving GHCi.
```

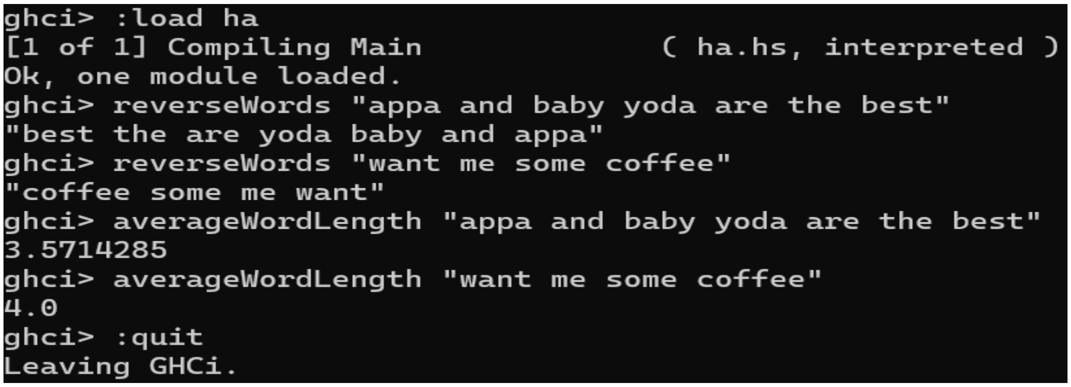
Task 3 - Puzzlers:

Code:



```haskell
reverseWords :: String -> String
reverseWords x = unwords (reverse (words x))

averageWordLength :: String -> Float
averageWordLength x = (fromIntegral (sum (map length (words x)))) / (fromIntegral (length (words x)))
```

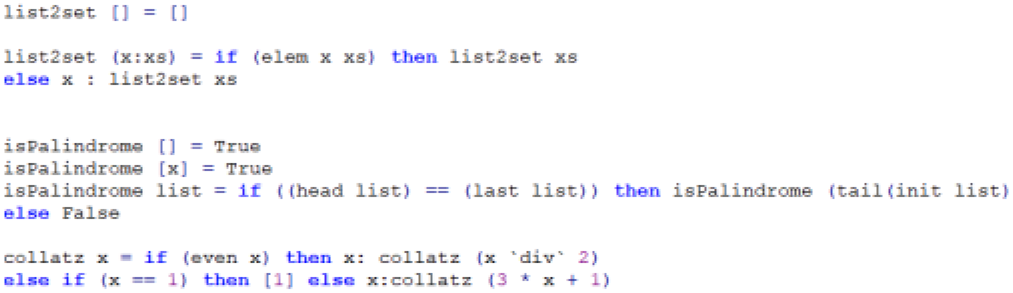
Demo:



```
ghci> :load ha
[1 of 1] Compiling Main             ( ha.hs, interpreted )
Ok, one module loaded.
ghci> reverseWords "appa and baby yoda are the best"
"best the are yoda baby and appa"
ghci> reverseWords "want me some coffee"
"coffee some me want"
ghci> averageWordLength "appa and baby yoda are the best"
3.5714285
ghci> averageWordLength "want me some coffee"
4.0
ghci> :quit
Leaving GHCi.
```

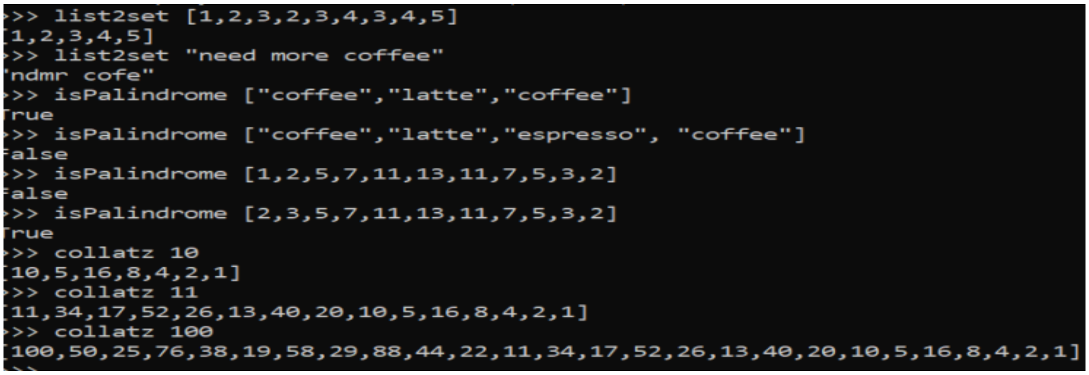
Task 4: Recursive List Processors:

Code:



```
list2set [] = []

list2set (x:xs) = if (elem x xs) then list2set xs
else x : list2set xs


isPalindrome [] = True
isPalindrome [x] = True
isPalindrome list = if ((head list) == (last list)) then isPalindrome (tail(init list)
else False

collatz x = if (even x) then x: collatz (x `div` 2)
else if (x == 1) then [1] else x:collatz (3 * x + 1)
```

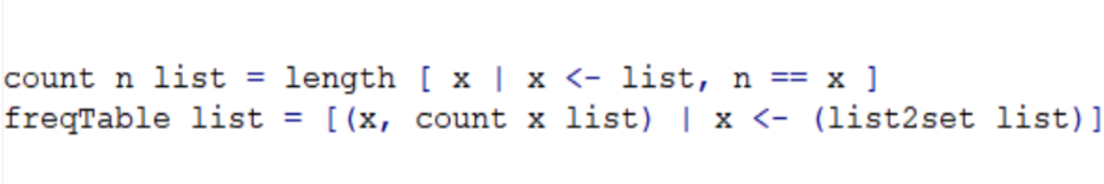
Demo:



```
>>> list2set [1,2,3,2,3,4,3,4,5]
[1,2,3,4,5]
>>> list2set "need more coffee"
"ndmr cofe"
>>> isPalindrome ["coffee","latte","coffee"]
True
>>> isPalindrome ["coffee","latte","espresso", "coffee"]
False
>>> isPalindrome [1,2,5,7,11,13,11,7,5,3,2]
False
>>> isPalindrome [2,3,5,7,11,13,11,7,5,3,2]
True
>>> collatz 10
[10,5,16,8,4,2,1]
>>> collatz 11
[11,34,17,52,26,13,40,20,10,5,16,8,4,2,1]
>>> collatz 100
[100,50,25,76,38,19,58,29,88,44,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1]
>>>
```

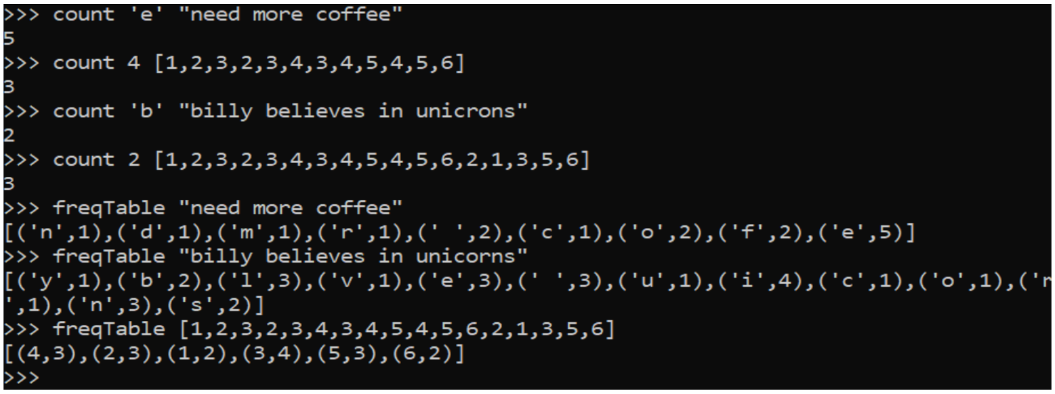
Task 5: List Comprehensions:

Code:



```
count n list = length [ x | x <- list, n == x ]
freqTable list = [(x, count x list) | x <- (list2set list)]
```

Demo:



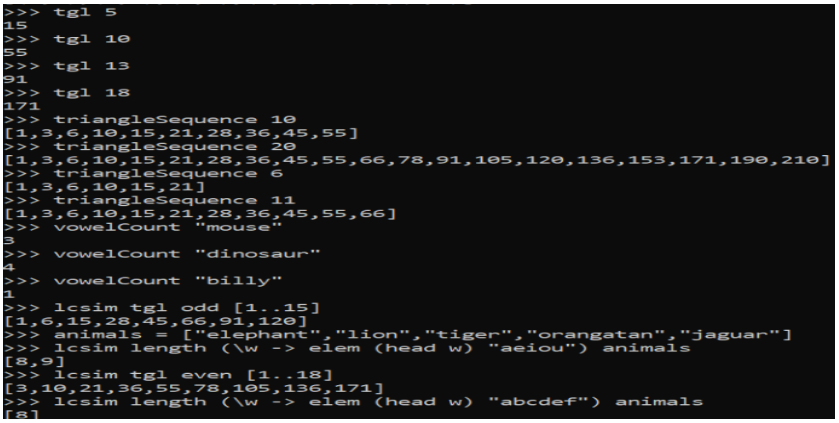
```
>>> count 'e' "need more coffee"
5
>>> count 4 [1,2,3,2,3,4,3,4,5,4,5,6]
3
>>> count 'b' "billy believes in unicrons"
2
>>> count 2 [1,2,3,2,3,4,3,4,5,4,5,6,2,1,3,5,6]
3
>>> freqTable "need more coffee"
[('n',1),('d',1),('m',1),('r',1),(' ',2),('c',1),('o',2),('f',2),('e',5)]
>>> freqTable "billy believes in unicorns"
[('y',1),('b',2),('l',3),('v',1),('e',3),(' ',3),('u',1),('i',4),('c',1),('o',1),('r
',1),('n',3),('s',2)]
>>> freqTable [1,2,3,2,3,4,3,4,5,4,5,6,2,1,3,5,6]
[(4,3),(2,3),(1,2),(3,4),(5,3),(6,2)]
>>>
```

Task 6: Higher Order Functions:

Code:

```
tgl x = foldl (+) 0 [1 .. x]
triangleSequence x = map tgl [1 .. x]
vowelCount lowString = length theVowelList
  where theVowelList = filter (\x -> elem x "a e i o u") lowString
lcsim fn pred list = map fn filList
  where filList = filter pred list
```

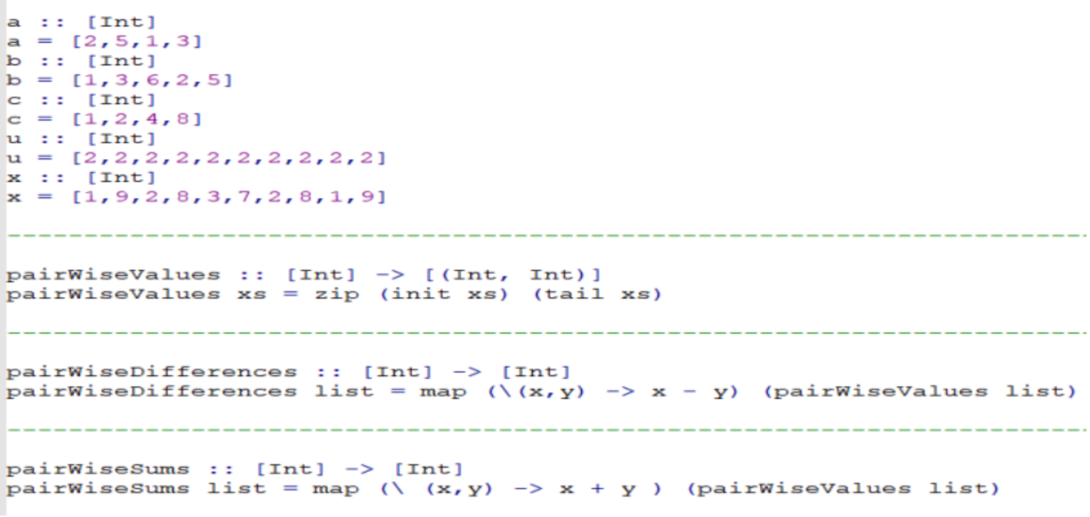
Demo:



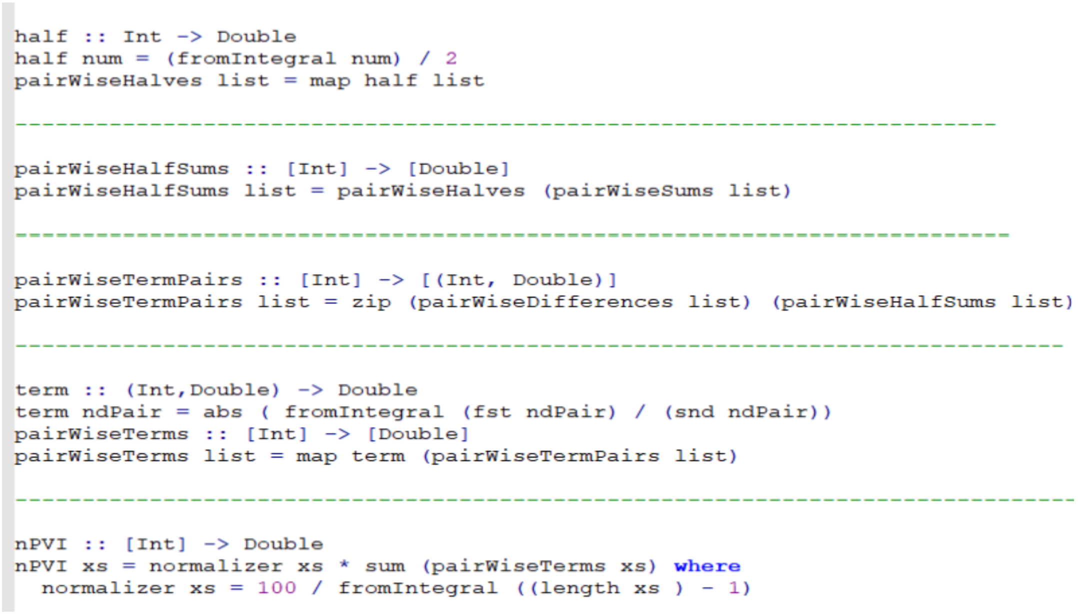
```
>>> tgl 5
15
>>> tgl 10
55
>>> tgl 13
91
>>> tgl 18
171
>>> triangleSequence 10
[1,3,6,10,15,21,28,36,45,55]
>>> triangleSequence 20
[1,3,6,10,15,21,28,36,45,55,66,78,91,105,120,136,153,171,190,210]
>>> triangleSequence 6
[1,3,6,10,15,21]
>>> triangleSequence 11
[1,3,6,10,15,21,28,36,45,55,66]
>>> vowelCount "mouse"
3
>>> vowelCount "dinosaur"
4
>>> vowelCount "billy"
1
>>> lcsim tgl odd [1..15]
[1,6,15,28,45,66,91,120]
>>> animals = ["elephant","lion","tiger","orangatan","jaguar"]
>>> lcsim length (\w -> elem (head w) "aeiou") animals
[8,9]
>>> lcsim tgl even [1..18]
[3,10,21,36,55,78,105,136,171]
>>> lcsim length (\w -> elem (head w) "abcdef") animals
[8]
```

Task 7: An Interestign Statistic: nPVI:

Code:



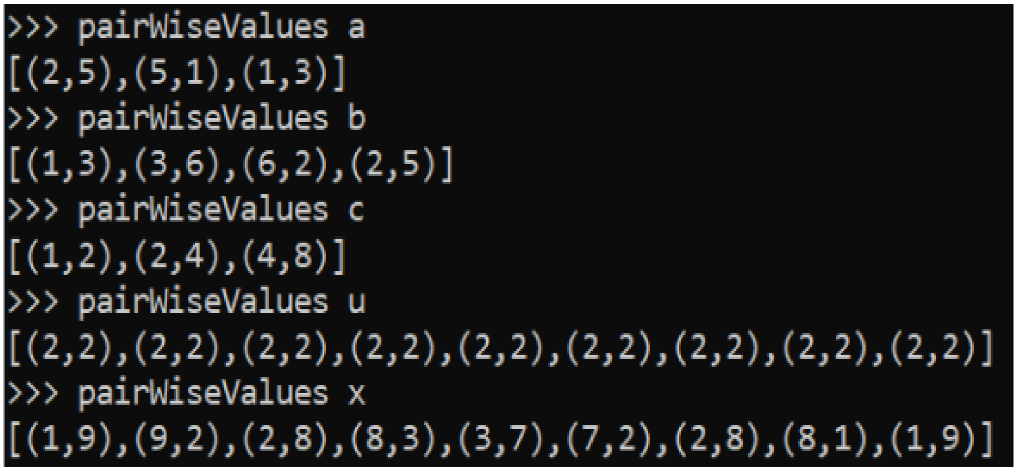
```
a :: [Int]
a = [2,5,1,3]
b :: [Int]
b = [1,3,6,2,5]
c :: [Int]
c = [1,2,4,8]
u :: [Int]
u = [2,2,2,2,2,2,2,2,2,2]
x :: [Int]
x = [1,9,2,8,3,7,2,8,1,9]

---------------------------------------------------------------------------------

pairWiseValues :: [Int] -> [(Int, Int)]
pairWiseValues xs = zip (init xs) (tail xs)

---------------------------------------------------------------------------------

pairWiseDifferences :: [Int] -> [Int]
pairWiseDifferences list = map (\(x,y) -> x - y) (pairWiseValues list)

---------------------------------------------------------------------------------

pairWiseSums :: [Int] -> [Int]
pairWiseSums list = map (\ (x,y) -> x + y ) (pairWiseValues list)
```



```
half :: Int -> Double
half num = (fromIntegral num) / 2
pairWiseHalves list = map half list

----------------------------------------------------------------------

pairWiseHalfSums :: [Int] -> [Double]
pairWiseHalfSums list = pairWiseHalves (pairWiseSums list)

-----------------------------------------------------------------------

pairWiseTermPairs :: [Int] -> [(Int, Double)]
pairWiseTermPairs list = zip (pairWiseDifferences list) (pairWiseHalfSums list)

---------------------------------------------------------------------------

term :: (Int,Double) -> Double
term ndPair = abs ( fromIntegral (fst ndPair) / (snd ndPair))
pairWiseTerms :: [Int] -> [Double]
pairWiseTerms list = map term (pairWiseTermPairs list)

----------------------------------------------------------------------------

nPVI :: [Int] -> Double
nPVI xs = normalizer xs * sum (pairWiseTerms xs) where
  normalizer xs = 100 / fromIntegral ((length xs ) - 1)
```

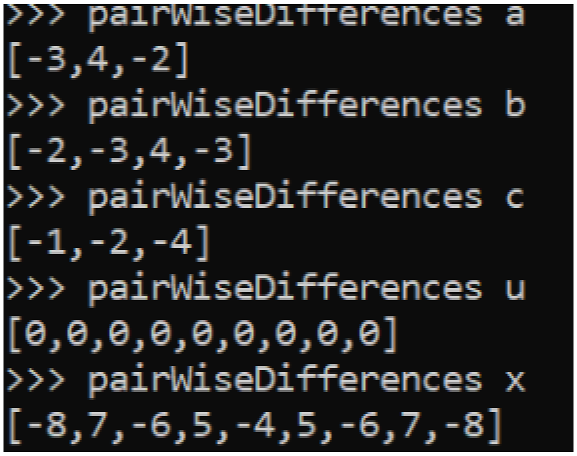
Task 7a-i: Demo:

7B)



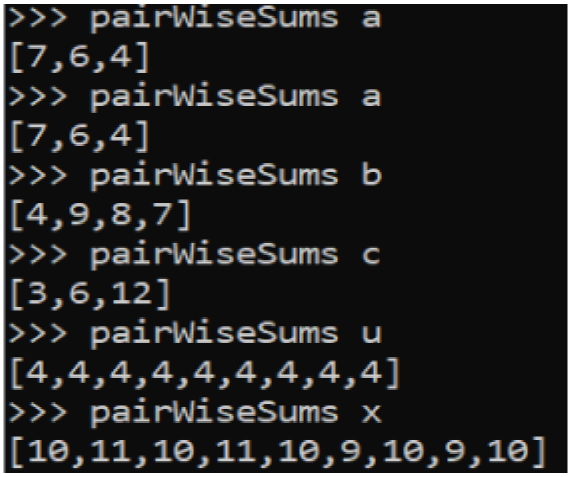
```
>>> pairWiseValues a
[(2,5),(5,1),(1,3)]
>>> pairWiseValues b
[(1,3),(3,6),(6,2),(2,5)]
>>> pairWiseValues c
[(1,2),(2,4),(4,8)]
>>> pairWiseValues u
[(2,2),(2,2),(2,2),(2,2),(2,2),(2,2),(2,2),(2,2),(2,2)]
>>> pairWiseValues x
[(1,9),(9,2),(2,8),(8,3),(3,7),(7,2),(2,8),(8,1),(1,9)]
```

7C)



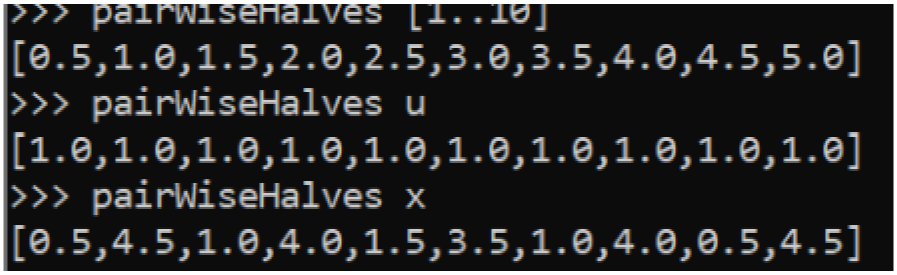
```
>>> pairWiseDifferences a
[-3,4,-2]
>>> pairWiseDifferences b
[-2,-3,4,-3]
>>> pairWiseDifferences c
[-1,-2,-4]
>>> pairWiseDifferences u
[0,0,0,0,0,0,0,0,0]
>>> pairWiseDifferences x
[-8,7,-6,5,-4,5,-6,7,-8]
```

7D)



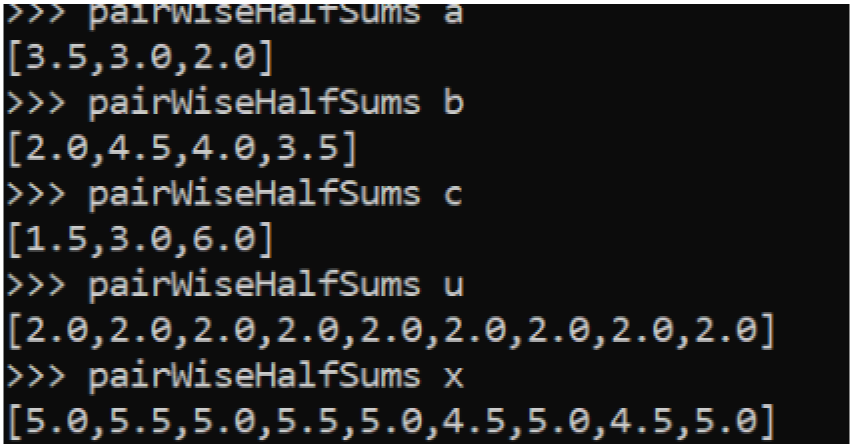
```
>>> pairWiseSums a
[7,6,4]
>>> pairWiseSums a
[7,6,4]
>>> pairWiseSums b
[4,9,8,7]
>>> pairWiseSums c
[3,6,12]
>>> pairWiseSums u
[4,4,4,4,4,4,4,4,4]
>>> pairWiseSums x
[10,11,10,11,10,9,10,9,10]
```

7E)



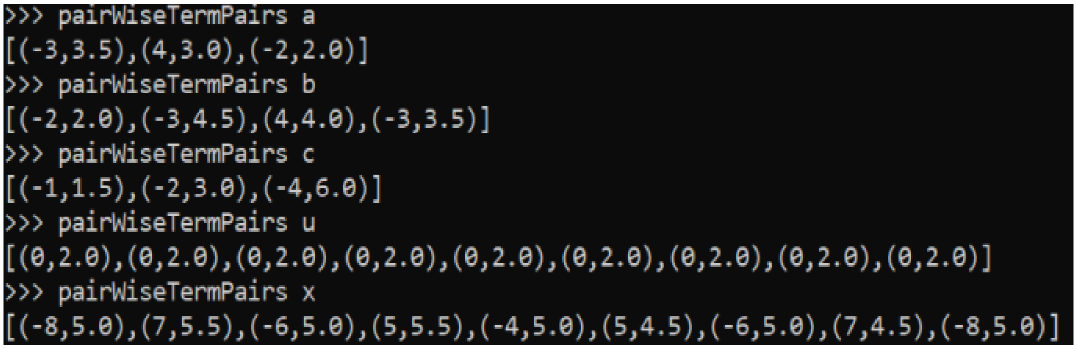
```
>>> pairWiseHalves [1..10]
[0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0]
>>> pairWiseHalves u
[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]
>>> pairWiseHalves x
[0.5,4.5,1.0,4.0,1.5,3.5,1.0,4.0,0.5,4.5]
```

7F)



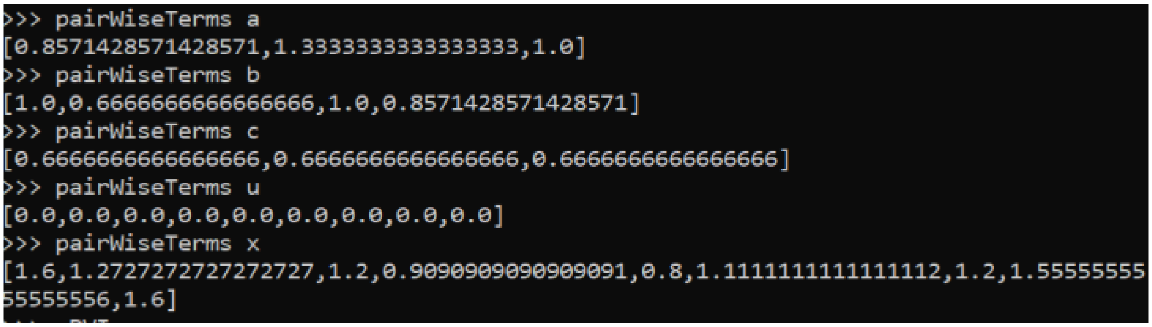
```
>>> pairWiseHalfSums a
[3.5,3.0,2.0]
>>> pairWiseHalfSums b
[2.0,4.5,4.0,3.5]
>>> pairWiseHalfSums c
[1.5,3.0,6.0]
>>> pairWiseHalfSums u
[2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0,2.0]
>>> pairWiseHalfSums x
[5.0,5.5,5.0,5.5,5.0,4.5,5.0,4.5,5.0]
```

7G)



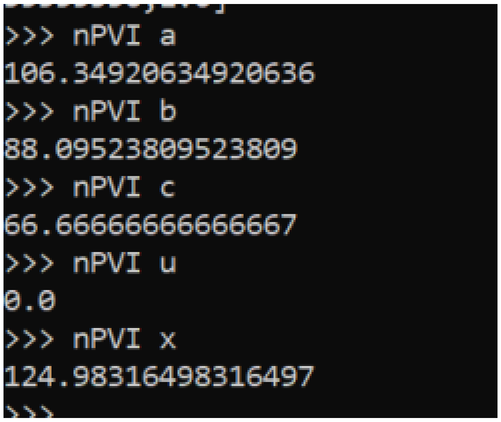
```
>>> pairWiseTermPairs a
[(-3,3.5),(4,3.0),(-2,2.0)]
>>> pairWiseTermPairs b
[(-2,2.0),(-3,4.5),(4,4.0),(-3,3.5)]
>>> pairWiseTermPairs c
[(-1,1.5),(-2,3.0),(-4,6.0)]
>>> pairWiseTermPairs u
[(0,2.0),(0,2.0),(0,2.0),(0,2.0),(0,2.0),(0,2.0),(0,2.0),(0,2.0),(0,2.0)]
>>> pairWiseTermPairs x
[(-8,5.0),(7,5.5),(-6,5.0),(5,5.5),(-4,5.0),(5,4.5),(-6,5.0),(7,4.5),(-8,5.0)]
```

7H)



```
>>> pairWiseTerms a
[0.8571428571428571,1.3333333333333333,1.0]
>>> pairWiseTerms b
[1.0,0.6666666666666666,1.0,0.8571428571428571]
>>> pairWiseTerms c
[0.6666666666666666,0.6666666666666666,0.6666666666666666]
>>> pairWiseTerms u
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]
>>> pairWiseTerms x
[1.6,1.2727272727272727,1.2,0.9090909090909091,0.8,1.1111111111111112,1.2,1.55555555
55555556,1.6]
>>> PWT
```

7I)

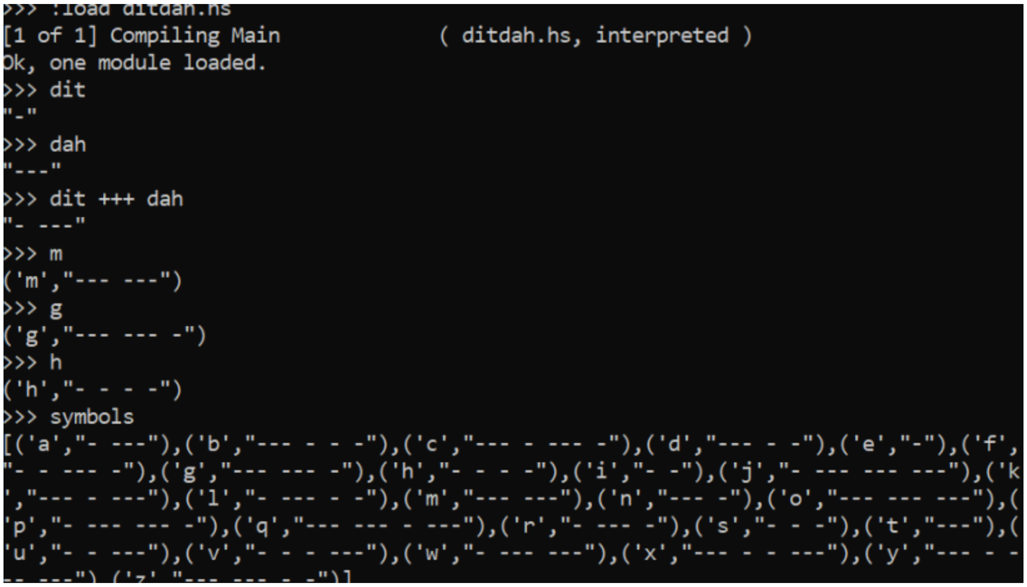


```
>>> nPVI a
106.34920634920636
>>> nPVI b
88.09523809523809
>>> nPVI c
66.66666666666667
>>> nPVI u
0.0
>>> nPVI x
124.98316498316497
>>>
```

Task 8: Historic Code: The Dit Dah Code:

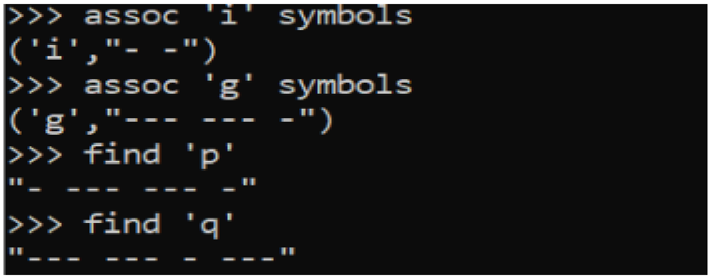
Demo (8a-d):

8A)



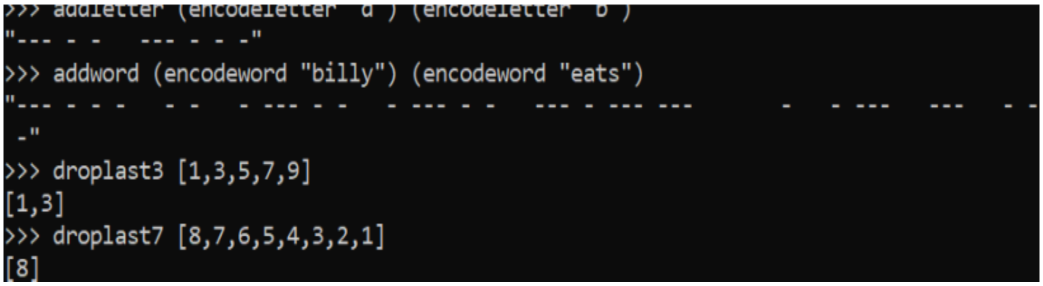
```
>>> :load ditdah.hs
[1 of 1] Compiling Main             ( ditdah.hs, interpreted )
Ok, one module loaded.
>>> dit
"-"
>>> dah
"---"
>>> dit +++ dah
"- ---"
>>> m
('m',"--- ---")
>>> g
('g',"--- --- -")
>>> h
('h',"- - - -")
>>> symbols
[('a',"- ---"),('b',"--- - - -"),('c',"--- - --- -"),('d',"--- - -"),('e',"-"),('f',
"- - --- -"),('g',"--- --- -"),('h',"- - - -"),('i',"- -"),('j',"- --- --- ---"),('k
',"--- - ---"),('l',"- --- - -"),('m',"--- ---"),('n',"--- -"),('o',"--- --- ---"),(
'p',"- --- --- -"),('q',"--- --- - ---"),('r',"- --- -"),('s',"- - -"),('t',"---"),(
'u',"- - ---"),('v',"- - - ---"),('w',"- --- ---"),('x',"--- - - ---"),('y',"--- - -
-- ---"),('z',"--- --- - -")]
```

8B)



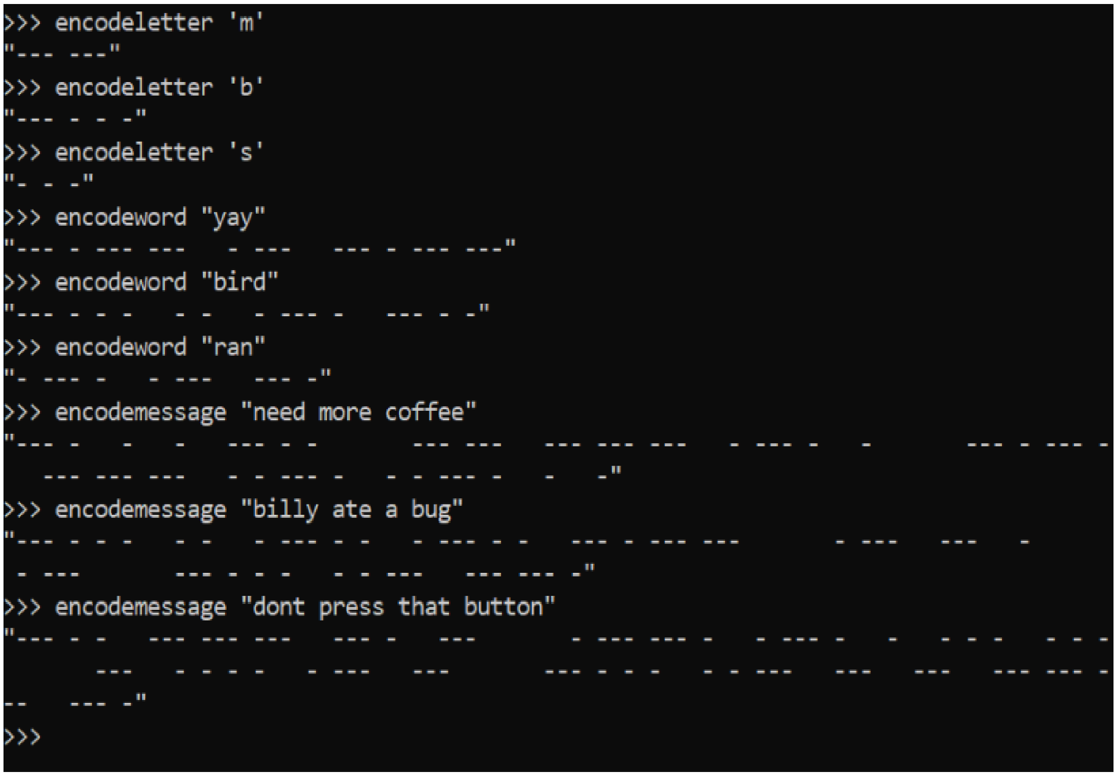
```
>>> assoc 'i' symbols
('i',"- -")
>>> assoc 'g' symbols
('g',"--- --- -")
>>> find 'p'
"- --- --- -"
>>> find 'q'
"--- --- - ---"
```

8C)



```
>>> addletter (encodeletter 'd') (encodeletter 'b')
"--- - -   --- - - -"
>>> addword (encodeword "billy") (encodeword "eats")
"--- - - -   - -   - --- - -   - --- - -   --- - --- ---       -   - ---   ---   - -
 -"
>>> droplast3 [1,3,5,7,9]
[1,3]
>>> droplast7 [8,7,6,5,4,3,2,1]
[8]
```

8D)



```
>>> encodeletter 'm'
"--- ---"
>>> encodeletter 'b'
"--- - - -"
>>> encodeletter 's'
"- - -"
>>> encodeword "yay"
"--- - --- ---   - ---   --- - --- ---"
>>> encodeword "bird"
"--- - - -   - -   - --- -   --- - -"
>>> encodeword "ran"
"- --- -   - ---   --- -"
>>> encodemessage "need more coffee"
"--- -   -   -   --- - -       --- ---   --- --- ---   - --- -   -       --- - --- -
   --- --- ---   - - --- -   - - --- -   -   -"
>>> encodemessage "billy ate a bug"
"--- - - -   - -   - --- - -   - --- - -   --- - --- ---       - ---   ---   -
 - ---       --- - - -   - - ---   --- --- -"
>>> encodemessage "dont press that button"
"--- - -   --- --- ---   --- -   ---       - --- --- -   - --- -   -   - - -   - - -
       ---   - - - -   - ---   ---       --- - - -   - - ---   ---   ---   --- --- -
--   --- -"
>>>
```