

19CSE313

Principles of Programming Languages

S Abhishek

AM.EN.U4CSE19147

```
Prelude Data.List> intersperse '.' "MONKEY"
"M.O.N.K.E.Y"
Prelude Data.List> intercalate " " ["hey", "there", "guys"]
"hey there guys"
Prelude Data.List> intercalate [0,0,0] [[1,2,3],[4,5,6],[7,8,9]]
[1,2,3,0,0,0,4,5,6,0,0,0,7,8,9]
Prelude Data.List> transpose [[1,2,3],[4,5,6],[7,8,9]]
[[1,4,7],[2,5,8],[3,6,9]]
Prelude Data.List> transpose ["hey", "there", "guys"]
["htg","ehu","yey","rs","e"]
Prelude Data.List> concat [[3,4,5],[2,3,4],[2,1,1]]
[3,4,5,2,3,4,2,1,1]
Prelude Data.List> and $ map (==4) [4,4,4,3,4]
False
Prelude Data.List> or $ map (==4) [4,4,4,3,4]
True
Prelude Data.List> any (==4) [4,4,4,3,4]
True
Prelude Data.List> all (==4) [4,4,4,3,4]
False
Prelude Data.List> all (`elem` ['A'..'Z']) "HEYGUYSwhatsup"
False
Prelude Data.List> any (`elem` ['A'..'Z']) "HEYGUYSwhatsup"
True
Prelude Data.List> take 10 $ iterate (*2) 1
[1,2,4,8,16,32,64,128,256,512]
Prelude Data.List> splitAt 3 "heyman"
("hey", "man")
Prelude Data.List> splitAt (-3) "heyman"
("", "heyman")
Prelude Data.List> takeWhile (>3) [6,5,3,2,1,2,3,4,5,4,3,2,1]
[6,5]
Prelude Data.List> takeWhile (/=' ') "This is a sentence"
"This"
Prelude Data.List> dropWhile (/=' ') "This is a sentence"
" is a sentence"
Prelude Data.List> sort [4,5,7,8,1,3,4,5,2,7,9,0]
[0,1,2,3,4,4,5,5,7,7,8,9]
Prelude Data.List> group [1,1,1,1,1,2,2,2,2,3,4,5,5,5,6,6,7]
[[1,1,1,1,1],[2,2,2,2],[3],[4],[5,5,5],[6,6],[7]]
```

```

Prelude Data.List> inits "w00t"
["","w","w0","w00","w00t"]
Prelude Data.List> tail "w00t"
"00t"
Prelude Data.List> tails "w00t"
["w00t","00t","0t","t",""]
Prelude Data.List> let w = "w00t" in zip (inits w) (tails w)
[("", "w00t"), ("w", "00t"), ("w0", "0t"), ("w00", "t"), ("w00t", "")]
Prelude Data.List> "cat" `isInfixOf` "im a cat burglar"
True
Prelude Data.List> "hey" `isPrefixOf` "hey there!"
True
Prelude Data.List> find (>4) [1,2,3,4,5,6]
Just 5
Prelude Data.List> find (>6) [1,2,3,4,5,6]
Nothing
Prelude Data.List> :t find
find :: Foldable t => (a -> Bool) -> t a -> Maybe a

```

```

Prelude Data.List> 4 `elemIndex` [1,2,3,4,5]
Just 3
Prelude Data.List> 6 `elemIndex` [1,2,3,4,5]
Nothing
Prelude Data.List> ' ' `elemIndices` "Where are the Spaces?"
[5,9,13]

```

```

Prelude Data.List> findIndex (==4) [5,4,3,3,2,4,4]
Just 1
Prelude Data.List> findIndices (==4) [5,4,3,3,2,4,4]
[1,5,6]
Prelude Data.List> findIndices (==6) [5,4,3,3,2,4,4]
[]
Prelude Data.List> findIndex (==6) [5,4,3,3,2,4,4]
Prelude Data.List> findIndices `elem` ['A'..'Z'] "Where are the Caps?"
[0,14]
Prelude Data.List> findIndices `elem` ['A'..'Z'] "Where Are The Caps?"
[0,6,10,14]
Prelude Data.List> delete 'h' "hey there"
"ey there"

```

```

Prelude Data.List> delete 'h' . delete 'h' $ "hey there"
"ey tere"
Prelude Data.List> [1..10] \\ [2,5,9]
[1,3,4,6,7,8,10]
Prelude Data.List> "Hey I am a baby" \\ "I am a"
"Hey  baby"
Prelude Data.List> "hey man" `union` "man what's up"
"hey manwt'sup"
Prelude Data.List> [1..7] `intersect` [5..10]
[5,6,7]
Prelude Data.List> insert 4 [3,5,1,2]
[3,4,5,1,2]

```

```
Prelude> import Data.Char
Prelude Data.Char> all isAlphaNum "bobby283"
True
Prelude Data.Char> all isAlphaNum "bobby"
True
Prelude Data.Char> all isAlphaNum "bobby "
False
Prelude Data.Char> generalCategory ' '
Space
Prelude Data.Char> generalCategory 'A'
UppercaseLetter
Prelude Data.Char> generalCategory '.'
OtherPunctuation
Prelude Data.Char> generalCategory 'a'
LowercaseLetter
```

```
Prelude Data.Char> generalCategory '4'
DecimalNumber
Prelude Data.Char> words "hey guys hello"
["hey", "guys", "hello"]
```

```
Prelude Data.Char> ord 'a'
97
Prelude Data.Char> chr 97
'a'
Prelude Data.Char> map ord "abcdefgh"
[97,98,99,100,101,102,103,104]
Prelude Data.Char> map digitToInt "34538"
[3,4,5,3,8]
Prelude Data.Char> map digitToInt "FF85AB"
[15,15,8,5,10,11]
Prelude Data.Char> intToDigit 15
'f'
Prelude Data.Char> intToDigit 5
'5'
```

```
Prelude Data.Char Data.List> groupBy (==) "hey guys hello"
["h","e","y"," ","g","u","y","s"," ","h","e","l","l","o"]
Prelude Data.Char Data.List> groupBy (<=) [1,2,2,3,1,2,0,4,5,2]
[[1,2,2,3,1,2],[0,4,5,2]]
```

Thankyou!!