Някои функции за работа със списъци, реализирани в Prelude.hs:

`:`	`a -> [a] -> [a]`	Add a single element to the front of a list. `1:[2,3] => [1,2,3]`
`++`	`[a] -> [a] -> [a]`	Join two lists together. `"ab"++"cde" => "abcde"`
`!!`	`[a] -> Int -> a`	`xs!!n` returns the nth element of `xs`, starting at the beginning and counting from 0. `[14,7,3]!!1 => 7`
`concat`	`[[a]] -> [a]`	Concatenate a list of lists into a single list. `concat [[2,3],[],[4]] => [2,3,4]`
`length`	`[a] -> Int`	The length of the list. `length "word" => 4`

head	[a] -> a	The first element of the list. head “word” => ‘w’
last	[a] -> a	The last element of the list. last “word” => ‘d’
tail	[a] -> [a]	All but the first element of the list. tail “word” => “ord”
init	[a] -> [a]	All but the last element of the list. init “word” => “wor”
replicate	Int -> a -> [a]	Make a list of n copies of the item. replicate 3 ‘c’ => “ccc”
take	Int -> [a] -> [a]	Take n elements from the front of a list. take 3 “Peccary” => “Pec”

drop	Int -> [a] -> [a]	Drop n elements from the front of a list. drop 3 “Peccary” => “cary”
splitAt	Int->[a]->([a],[a])	Split a list at a given position. splitAt 3 “Peccary” => (“Pec”,”cary”)
reverse	[a] -> [a]	Reverse the order of the elements. reverse [1,2,3] => [3,2,1]
zip	[a]->[b]->[(a,b)]	Take a pair of lists into a list of pairs. zip [1,2] [3,4,5] => [(1,3),(2,4)]

unzip	[(a,b)] -> ([a],[b])	Take a list of pairs into a pair of lists. unzip [(1,5),(2,6)] => ([1,2],[5,6])
and	[Bool] -> Bool	The conjunction of a list of Booleans. and [True,False] => False
or	[Bool] -> Bool	The disjunction of a list of Booleans. or [True,False] => True
sum	[Int] -> Int [Float] -> Float	The sum of a numeric list. sum [2,3,4] => 9
product	[Int] -> Int [Float] -> Float	The product of a numeric list. product [0.1,0.4 .. 1] => 0.028