```
ouzo2:code$ ghci
GHCi, version 8.2.2: http://www.haskell.org/ghc/  :? for help
Loaded GHCi configuration from /Users/hallgren/.ghci
Prelude> :l WorkingWithLists.hs
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> null []
True
*Main> null [1,2,3]
False
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> [1,2,3]++[4,5,6]
[1,2,3,4,5,6]
*Main> "abc"++"def"
"abcdef"
*Main> :t null
null :: [a] -> Bool
*Main> :t ++

<interactive>:1:1: error: parse error on input ‘++’
*Main> :t (++)
(++) :: [a] -> [a] -> [a]
*Main> (++) "abc" "def"
"abcdef"
*Main> [1..10]
[1,2,3,4,5,6,7,8,9,10]
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> snoc "abc" 'd'
"abcd"
*Main> reverse [1..100]
[100,99,98,97,96,95,94,93,92,91,90,89,88,87,86,85,84,83,82,81,80,79,78,77,76,75,
74,73,72,71,70,69,68,67,66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,48
,47,46,45,44,43,42,41,40,39,38,37,36,35,34,33,32,31,30,29,28,27,26,25,24,23,22,2
1,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]
*Main> reverse [1..1000]
[1000,999,998,997,996,995,994,993,992,991,990,989,988,987,986,985,984,983,982,98
1,980,979,978,977,976,975,974,973,972,971,970,969,968,967,966,965,964,963,962,96
1,960,959,958,957,956,955,954,953,952,951,950,949,948,947,946,945,944,943,942,94
1,940,939,938,937,936,935,934,933,932,931,930,929,928,927,926,925,924,923,922,92
1,920,919,918,917,916,915,914,913,912,911,910,909,908,907,906,905,904,903,902,90
1,900,899,898,897,896,895,894,893,892,891,890,889,888,887,886,885,884,883,882,88
1,880,879,878,877,876,875,874,873,872,871,870,869,868,867,866,865,864,863,862,86
1,860,859,858,857,856,855,854,853,852,851,850,849,848,847,846,845,844,843,842,84
1,840,839,838,837,836,835,834,833,832,831,830,829,828,827,826,825,824,823,822,82
1,820,819,818,817,816,815,814,813,812,811,810,809,808,807,806,805,804,803,802,80
1,800,799,798,797,796,795,794,793,792,791,790,789,788,787,786,785,784,783,782,78
1,780,779,778,777,776,775,774,773,772,771,770,769,768,767,766,765,764,763,762,76
1,760,759,758,757,756,755,754,753,752,751,750,749,748,747,746,745,744,743,742,74
1,740,739,738,737,736,735,734,733,732,731,730,729,728,727,726,725,724,723,722,72
1,720,719,718,717,716,715,714,713,712,711,710,709,708,707,706,705,704,703,702,70
1,700,699,698,697,696,695,694,693,692,691,690,689,688,687,686,685,684,683,682,68
1,680,679,678,677,676,675,674,673,672,671,670,669,668,667,666,665,664,663,662,66
1,660,659,658,657,656,655,654,653,652,651,650,649,648,647,646,645,644,643,642,64
1,640,639,638,637,636,635,634,633,632,631,630,629,628,627,626,625,624,623,622,62
1,620,619,618,617,616,615,614,613,612,611,610,609,608,607,606,605,604,603,602,60
1,600,599,598,597,596,595,594,593,592,591,590,589,588,587,586,585,584,583,582,58
1,580,579,578,577,576,575,574,573,572,571,570,569,568,567,566,565,564,563,562,56
1,560,559,558,557,556,555,554,553,552,551,550,549,548,547,546,545,544,543,542,54
1,540,539,538,537,536,535,534,533,532,531,530,529,528,527,526,525,524,523,522,52
1,520,519,518,517,516,515,514,513,512,511,510,509,508,507,506,505,504,503,502,50
1,500,499,498,497,496,495,494,493,492,491,490,489,488,487,486,485,484,483,482,48
1,480,479,478,477,476,475,474,473,472,471,470,469,468,467,466,465,464,463,462,46
1,460,459,458,457,456,455,454,453,452,451,450,449,448,447,446,445,444,443,442,44
1,440,439,438,437,436,435,434,433,432,431,430,429,428,427,426,425,424,423,422,42
1,420,419,418,417,416,415,414,413,412,411,410,409,408,407,406,405,404,403,402,40
1,400,399,398,397,396,395,394,393,392,391,390,389,388,387,386,385,384,383,382,38
1,380,379,378,377,376,375,374,373,372,371,370,369,368,367,366,365,364,363,362,36
1,360,359,358,357,356,355,354,353,352,351,350,349,348,347,346,345,344,343,342,34
1,340,339,338,337,336,335,334,333,332,331,330,329,328,327,326,325,324,323,322,32
1,320,319,318,317,316,315,314,313,312,311,310,309,308,307,306,305,304,303,302,30
1,300,299,298,297,296,295,294,293,292,291,290,289,288,287,286,285,284,283,282,28
1,280,279,278,277,276,275,274,273,272,271,270,269,268,267,266,265,264,263,262,26
1,260,259,258,257,256,255,254,253,252,251,250,249,248,247,246,245,244,243,242,24
1,240,239,238,237,236,235,234,233,232,231,230,229,228,227,226,225,224,223,222,22
1,220,219,218,217,216,215,214,213,212,211,210,209,208,207,206,205,204,203,202,20
1,200,199,198,197,196,195,194,193,192,191,190,189,188,187,186,185,184,183,182,18
1,180,179,178,177,176,175,174,173,172,171,170,169,168,167,166,165,164,163,162,16
1,160,159,158,157,156,155,154,153,152,151,150,149,148,147,146,145,144,143,142,14
1,140,139,138,137,136,135,134,133,132,131,130,129,128,127,126,125,124,123,122,12
1,120,119,118,117,116,115,114,113,112,111,110,109,108,107,106,105,104,103,102,10
1,100,99,98,97,96,95,94,93,92,91,90,89,88,87,86,85,84,83,82,81,80,79,78,77,76,75
,74,73,72,71,70,69,68,67,66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,4
8,47,46,45,44,43,42,41,40,39,38,37,36,35,34,33,32,31,30,29,28,27,26,25,24,23,22,
21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]
*Main> sum (reverse [1..1000])
500500
*Main> sum (reverse [1..10000])
50005000
*Main> sum ([1..10000])
50005000
*Main> :set +s
*Main> sum ([1..10000])
50005000
(0.01 secs, 1,675,896 bytes)
*Main> sum (reverse [1..10000])
50005000
(6.77 secs, 8,760,046,488 bytes)
*Main> Prelude.take 5 [10..20]
[10,11,12,13,14]
(0.01 secs, 76,368 bytes)
*Main> Prelude.take 0 [10..20]
[]
(0.01 secs, 60,080 bytes)
*Main> Prelude.take 100 [10..20]
```

```
[10,11,12,13,14,15,16,17,18,19,20]
(0.01 secs, 84,712 bytes)
*Main> Prelude.take (-1) [10..20]
[]
(0.01 secs, 60,224 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> take 0 [1..10]
[]
(0.01 secs, 64,232 bytes)
*Main> take 5 [1..10]
[1,2,3,4,5]
(0.00 secs, 69,312 bytes)
*Main> take 20 [1..10]
[1,2,3,4,5,6,7,8,9,10]
(0.01 secs, 78,136 bytes)
*Main> take (-2) [1..10]
[1,2,3,4,5,6,7,8,9,10]
(0.01 secs, 82,560 bytes)
*Main> Prelude.take (-2) [1..10]
[]
(0.01 secs, 64,336 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_take
*** Failed! Falsifiable (after 2 tests and 1 shrink):
-1
[]
(0.09 secs, 272,136 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_take
+++ OK, passed 100 tests.
(0.01 secs, 5,613,544 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> drop 3 "abcdef"
"def"
(0.00 secs, 64,728 bytes)
*Main> drop 0 "abcdef"
"abcdef"
(0.00 secs, 64,248 bytes)
*Main> drop (-1) "abcdef"
"abcdef"
(0.00 secs, 68,432 bytes)
*Main> drop 10 "abcdef"
""
(0.00 secs, 65,216 bytes)
*Main> drop 6 "abcdef"
""
```

```
(0.00 secs, 61,280 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_take_drop
+++ OK, passed 100 tests.
(0.01 secs, 2,922,104 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck no
noShrinking        nonprop_take_drop  not                   notElem
*Main> quickCheck nonprop_take_drop
+++ OK, passed 100 tests.
(0.01 secs, 2,997,384 bytes)
*Main> prop_take_drop 5 "abcdefg"
True
(0.00 secs, 67,264 bytes)
*Main> nonprop_take_drop 5 "abcdefg"
False
(0.00 secs, 65,392 bytes)
*Main> veCheck nonprop_take_drop
vector                  verboseCheck            verboseCheckWith
vectorOf                verboseCheckAll         verboseCheckWithResult
verbose                 verboseCheckResult
*Main> verboseCheck nonprop_take_drop
Passed:
0
[]

Passed:
-1
[()]

Passed:
-2
[]

Passed:
2
[(),()]

Passed:
-1
[(),(),()]

Passed:
-1
[]

Passed:
1
[(),(),(),(),(),()]
```

```
Passed:
7
[(),(),()]

Passed:
5
[(),(),(),(),(),()]

Passed:
-9
[(),(),(),(),(),(),(),(),()]

Passed:
-3
[(),(),(),(),(),()]

Passed:
5
[()]

Passed:
-7
[(),(),(),(),(),(),(),(),()]

Passed:
10
[(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
7
[(),()]

Passed:
15
[(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
13
[(),(),(),(),(),(),(),(),()]

Passed:
2
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-2
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-7
[(),(),()]

Passed:
-8
```

```
[(),(),()]

Passed:
-9
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
12
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-4
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-3
[(),(),(),(),()]

Passed:
-23
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-6
[(),(),(),(),(),(),()]

Passed:
-3
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
18
[(),(),(),(),(),(),(),(),(),(),()]

Passed:
1
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-26
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
19
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),()]

Passed:
20
[(),(),(),(),()]

Passed:
12
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]
```

```
Passed:
-29
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),()]

Passed:
-25
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),()]

Passed:
-5
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),()]

Passed:
-20
[(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
34
[(),(),(),(),(),()]

Passed:
26
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),()]

Passed:
15
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-39
[(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-11
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),()]

Passed:
-27
[(),(),(),(),()]

Passed:
-18
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-31
[(),()]
```

```
Passed:
-16
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-21
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
46
[(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
28
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),()]

Passed:
-49
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
34
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
)]

Passed:
-25
[(),(),(),(),()]

Passed:
-25
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),()]

Passed:
-27
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
32
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),()]

Passed:
-6
[(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-32
```

```
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),()]

Passed:
47
[(),(),(),(),(),(),(),()]

Passed:
53
[(),(),(),(),(),(),(),(),()]

Passed:
-18
[]

Passed:
55
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-45
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),()]

Passed:
-31
[]

Passed:
-45
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-63
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
40
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
62
[(),(),(),(),(),(),(),()]

Passed:
-31
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
58
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-52
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
2
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),(),(),()]

Passed:
-22
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
64
[(),(),()]

Passed:
-66
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),()]

Passed:
2
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
58
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-50
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),()]

Passed:
65
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
5
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),()]

Passed:
6
```

```
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-39
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-16
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),()]

Passed:
-62
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
0
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),()]

Passed:
77
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),()]

Passed:
26
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
11
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),()]

Passed:
-63
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
84
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),()]

Passed:
-71
[(),(),(),(),(),(),(),(),(),(),(),(),(),()]
```

```
Passed:
75
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
14
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-53
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
61
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-35
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),
(),(),(),(),(),(),(),(),()]

Passed:
-61
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
-2
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(
),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

Passed:
24
[(),(),(),(),()]

Passed:
69
[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),()]

+++ OK, passed 100 tests.
(0.06 secs, 11,133,576 bytes)
*Main> :t prop
prop_take       prop_take_drop  properFraction  property
*Main> :t prop_take_drop
prop_take_drop :: Eq a => Int -> [a] -> Bool
*Main> :t nonprop_take_drop
nonprop_take_drop :: Eq a => Int -> [a] -> Bool
*Main> :set -XNoE
-XNoEmptyCase                  -XNoExplicitForAll
```

```
-XNoEmptyDataDecls               -XNoExplicitNamespaces
-XNoExistentialQuantification    -XNoExtendedDefaultRules
*Main> :set -XNoEx
-XNoExistentialQuantification    -XNoExplicitNamespaces
-XNoExplicitForAll               -XNoExtendedDefaultRules
*Main> :set -XNoExtendedDefaultRules
*Main> quickCheck nonprop_take_drop

<interactive>:52:1: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘quickCheck’
      prevents the constraint ‘(Arbitrary a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Arbitrary a, Arbitrary b) => Arbitrary (Either a b)
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Ordering
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Integer
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        ...plus 19 others
        ...plus 62 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the expression: quickCheck nonprop_take_drop
      In an equation for ‘it’: it = quickCheck nonprop_take_drop

<interactive>:52:12: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘nonprop_take_drop’
      prevents the constraint ‘(Eq a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Eq b, Eq a) => Eq (Either a b)
          -- Defined in ‘Data.Either’
        instance Eq Ordering -- Defined in ‘GHC.Classes’
        instance Eq Integer
          -- Defined in ‘integer-gmp-1.0.1.0:GHC.Integer.Type’
        ...plus 23 others
        ...plus 80 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the first argument of ‘quickCheck’, namely ‘nonprop_take_drop’
      In the expression: quickCheck nonprop_take_drop
      In an equation for ‘it’: it = quickCheck nonprop_take_drop
(0.02 secs,)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck nonprop_take_drop
*** Failed! Falsifiable (after 4 tests and 3 shrinks):
1
[0,1]
(0.01 secs, 690,776 bytes)
*Main> quickCheck prop_take_drop
+++ OK, passed 100 tests.
(0.01 secs, 5,747,424 bytes)
*Main> :r
```

```
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck nonprop_take_drop
*** Failed! Falsifiable (after 14 tests and 7 shrinks):
1
[True,False]
(0.01 secs, 1,147,048 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> zip [1..10] [21.30]
[(1,21.3)]
(0.01 secs, 73,192 bytes)
*Main> zip [1..10] [21..30]
[(1,21),(2,22),(3,23),(4,24),(5,25),(6,26),(7,27),(8,28),(9,29),(10,30)]
(0.01 secs, 115,152 bytes)
*Main> zip [1..10] [21..25]
[(1,21),(2,22),(3,23),(4,24),(5,25)]
(0.01 secs, 89,656 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> zip [1..10] [21..25]
[(1,21),(2,22),(3,23),(4,24),(5,25)]
(0.01 secs, 90,024 bytes)
*Main> unzip (zip [1..10] [21..25])
([1,2,3,4,5],[21,22,23,24,25])
(0.01 secs, 86,720 bytes)
*Main> unzip (zip [1..10] [21..30])
([1,2,3,4,5,6,7,8,9,10],[21,22,23,24,25,26,27,28,29,30])
(0.01 secs, 108,400 bytes)
*Main> :t fst
fst :: (a, b) -> a
*Main> :t snd
snd :: (a, b) -> b
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )

WorkingWithLists.hs:81:32: error:
    parse error on input ‘=’
    Perhaps you need a 'let' in a 'do' block?
    e.g. 'let x = 5' instead of 'x = 5'
   |
81 | prop_zip_unzip xys = zip xs ys = xys
   |                                ^
Failed, no modules loaded.
Prelude> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop
prop_take       prop_take_drop  prop_zip_unzip  properFraction  property
```

```
*Main> quickCheck prop_zip_unzip

<interactive>:71:1: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘quickCheck’
      prevents the constraint ‘(Arbitrary a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Arbitrary a, Arbitrary b) => Arbitrary (Either a b)
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Ordering
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Integer
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        ...plus 19 others
        ...plus 62 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the expression: quickCheck prop_zip_unzip
      In an equation for ‘it’: it = quickCheck prop_zip_unzip

<interactive>:71:12: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘prop_zip_unzip’
      prevents the constraint ‘(Eq a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Eq b, Eq a) => Eq (Either a b)
          -- Defined in ‘Data.Either’
        instance Eq Ordering -- Defined in ‘GHC.Classes’
        instance Eq Integer
          -- Defined in ‘integer-gmp-1.0.1.0:GHC.Integer.Type’
        ...plus 23 others
        ...plus 80 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the first argument of ‘quickCheck’, namely ‘prop_zip_unzip’
      In the expression: quickCheck prop_zip_unzip
      In an equation for ‘it’: it = quickCheck prop_zip_unzip
(0.02 secs,)
*Main> :t prop_zip_unzip
prop_zip_unzip :: (Num a, Eq a) => [(a, a)] -> Bool
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> :t prop_zip_unzip
prop_zip_unzip :: [(Bool, Int)] -> Bool
*Main> quickCheck prop_zip_unzip
+++ OK, passed 100 tests.
(0.02 secs, 8,741,528 bytes)
*Main> :R
unknown command ':R'
use :? for help.
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> :t prop_unzip_zip
prop_unzip_zip :: (Eq a1, Eq a2) => [a1] -> [a2] -> Bool
```

```
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_unzip_zip
+++ OK, passed 100 tests.
(0.01 secs, 8,325,688 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> qsort "Haskell"
"Haeklls"
(0.01 secs, 78,896 bytes)
*Main> qsort [100,99..1]
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,3
0,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,
57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83
,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100]
(0.02 secs, 2,543,232 bytes)
*Main> qsort [100]
[100]
(0.01 secs, 65,120 bytes)
*Main> qsort [100,5,9]
[5,9,100]
(0.01 secs, 71,736 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_qsort
+++ OK, passed 100 tests.
(0.03 secs, 15,083,848 bytes)
*Main> quickCheck prop_qsort
+++ OK, passed 100 tests.
(0.03 secs, 19,632,040 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_insert

<interactive>:90:1: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘quickCheck’
      prevents the constraint ‘(Arbitrary a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Arbitrary a, Arbitrary b) => Arbitrary (Either a b)
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Ordering
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        instance Arbitrary Integer
          -- Defined in ‘Test.QuickCheck.Arbitrary’
        ...plus 19 others
        ...plus 62 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the expression: quickCheck prop_insert
      In an equation for ‘it’: it = quickCheck prop_insert
```

```
<interactive>:90:12: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘prop_insert’
      prevents the constraint ‘(Ord a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance (Ord b, Ord a) => Ord (Either a b)
          -- Defined in ‘Data.Either’
        instance Ord Ordering -- Defined in ‘GHC.Classes’
        instance Ord Integer
          -- Defined in ‘integer-gmp-1.0.1.0:GHC.Integer.Type’
        ...plus 23 others
        ...plus 96 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In the first argument of ‘quickCheck’, namely ‘prop_insert’
      In the expression: quickCheck prop_insert
      In an equation for ‘it’: it = quickCheck prop_insert
(0.02 secs,)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_insert
*** Failed! Falsifiable (after 5 tests and 2 shrinks):
0
[1,0]
(0.01 secs, 470,488 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )

WorkingWithLists.hs:108:20: error:
    • Couldn't match expected type ‘Bool’ with actual type ‘Property’
    • In the expression: isSorted xs ==> isSorted (insert x xs)
      In an equation for ‘prop_insert’:
          prop_insert x xs = isSorted xs ==> isSorted (insert x xs)
    |
108 | prop_insert x xs = isSorted xs ==> isSorted (insert x xs)
    |                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Failed, no modules loaded.
Prelude> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> :t prop_insert
prop_insert :: Ord a => a -> [a] -> Property
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_insert
*** Gave up! Passed only 72 tests.
(0.04 secs, 25,422,792 bytes)
*Main>  stdArgs
Args {replay = Nothing, maxSuccess = 100, maxDiscardRatio = 10, maxSize = 100, c
hatty = True, maxShrinks = 9223372036854775807}
(0.00 secs, 124,720 bytes)
*Main> qu stdArgs
```

```
quickCheck            quickCheckWith        quotRem
quickCheckAll         quickCheckWithResult
quickCheckResult      quot
*Main> quickCheck stdArgs{maxDiscardRatio =100} prop_insert

<interactive>:99:1: error:
    • Couldn't match expected type ‘(Int -> [Int] -> Property) -> t’
                  with actual type ‘IO ()’
    • The function ‘quickCheck’ is applied to two arguments,
      but its type ‘Args -> IO ()’ has only one
      In the expression:
        quickCheck stdArgs {maxDiscardRatio = 100} prop_insert
      In an equation for ‘it’:
          it = quickCheck stdArgs {maxDiscardRatio = 100} prop_insert
    • Relevant bindings include it :: t (bound at <interactive>:99:1)
(0.01 secs,)
*Main> quickCheckWith stdArgs{maxDiscardRatio =100} prop_insert
+++ OK, passed 100 tests.
(0.07 secs, 54,207,888 bytes)
*Main> :t sh
show            showString      shrink1         shrinkList      shrinkRealFrac
showChar        shows           shrink2         shrinkMap       shrinkState
showList        showsPrec       shrinkInit      shrinkMapBy     shrinking
showParen       shrink          shrinkIntegral  shrinkNothing   shuffle
*Main> :t shrink
shrink :: Arbitrary a => a -> [a]
*Main> shrink 52

<interactive>:102:1: error:
    • Ambiguous type variable ‘a0’ arising from a use of ‘print’
      prevents the constraint ‘(Show a0)’ from being solved.
      Probable fix: use a type annotation to specify what ‘a0’ should be.
      These potential instances exist:
        instance [safe] Show Args -- Defined in ‘Test.QuickCheck.Test’
        instance [safe] Show Result -- Defined in ‘Test.QuickCheck.Test’
        instance (Show b, Show a) => Show (Either a b)
          -- Defined in ‘Data.Either’
        ...plus 25 others
        ...plus 115 instances involving out-of-scope types
        (use -fprint-potential-instances to see them all)
    • In a stmt of an interactive GHCi command: print it
(0.01 secs,)
*Main> shrink (52::Int)
[0,26,39,46,49,51]
(0.00 secs, 76,280 bytes)
*Main> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )

WorkingWithLists.hs:53:23: error:
    • Couldn't match expected type ‘Bool’ with actual type ‘Property’
    • In the expression:
        classify
          (n <= 0 || n > length xs) "extreme" (take n xs ++ drop n xs == xs)
      In an equation for ‘prop_take_drop’:
         prop_take_drop n xs
           = classify
               (n <= 0 || n > length xs) "extreme" (take n xs ++ drop n xs == x
s)
   |
53 | prop_take_drop n xs = classify (n<=0 || n>length xs) "extreme"
   |                       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^...
Failed, no modules loaded.
Prelude> :r
[1 of 1] Compiling Main             ( WorkingWithLists.hs, interpreted )
Ok, one module loaded.
*Main> quickCheck prop_take_drop
+++ OK, passed 100 tests (75% extreme).
(0.01 secs, 5,118,456 bytes)
*Main> list = [1..10]
(0.00 secs, 0 bytes)
*Main> square x = x*x
(0.00 secs, 0 bytes)
*Main> [square x|x<-list]
[1,4,9,16,25,36,49,64,81,100]
(0.01 secs, 85,928 bytes)
*Main> map square list
[1,4,9,16,25,36,49,64,81,100]
(0.01 secs, 80,560 bytes)
*Main> [x|x<-li
liftArbitrary   liftShrink      lines           listOf
liftArbitrary2  liftShrink2     list            listOf1
*Main> [x|x<-list,odd x]
[1,3,5,7,9]
(0.01 secs, 72,912 bytes)
*Main> filter odd list
[1,3,5,7,9]
(0.01 secs, 68,568 bytes)
*Main> [(x,y)<-x<-list,y<-list]

<interactive>:113:7: error:
    parse error on input ‘<-’
    Perhaps this statement should be within a 'do' block?
(0.00 secs,)
*Main> [(x,y)|x<-list,y<-list]
[(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(1,10),(2,1),(2,2),(2,3),
(2,4),(2,5),(2,6),(2,7),(2,8),(2,9),(2,10),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(
3,7),(3,8),(3,9),(3,10),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7),(4,8),(4,9),(4
,10),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7),(5,8),(5,9),(5,10),(6,1),(6,2),(6
,3),(6,4),(6,5),(6,6),(6,7),(6,8),(6,9),(6,10),(7,1),(7,2),(7,3),(7,4),(7,5),(7,
6),(7,7),(7,8),(7,9),(7,10),(8,1),(8,2),(8,3),(8,4),(8,5),(8,6),(8,7),(8,8),(8,9
),(8,10),(9,1),(9,2),(9,3),(9,4),(9,5),(9,6),(9,7),(9,8),(9,9),(9,10),(10,1),(10
,2),(10,3),(10,4),(10,5),(10,6),(10,7),(10,8),(10,9),(10,10)]
(0.01 secs, 550,880 bytes)
*Main> :set -XEx
-XExistentialQuantification  -XExplicitNamespaces
-XExplicitForAll             -XExtendedDefaultRules
*Main> :set -XExtendedDefaultRules
*Main> :t take
```