

Test Preview**TestSummary.txt: 1/1 Adithya Narayanan - anb122:j1:24**

```
1: Test Preview: Summary for anb122 of j1
2: PPT 24
3: -----
4:
5:   Public Tests:
6:     student-tests/tests/maxOf2:      3 / 3
7:     student-tests/tests/maxOf3:      3 / 3
8:     student-tests/tests/isADigit:    2 / 2
9:     student-tests/tests/isAlpha:     2 / 2
10:    student-tests/tests/digitToInt:   2 / 2
11:    student-tests/tests/toUpper:      2 / 2
12:    student-tests/tests/arithmeticSeq: 4 / 4
13:    student-tests/tests/geometricSeq:  4 / 4
14:    student-tests/tests/arithmeticSeries: 4 / 4
15:    student-tests/tests/geometricSeries: 4 / 4
16:    original-tests/tests/maxOf2:      3 / 3
17:    original-tests/tests/maxOf3:      3 / 3
18:    original-tests/tests/isADigit:    2 / 2
19:    original-tests/tests/isAlpha:     2 / 2
20:    original-tests/tests/digitToInt:   2 / 2
21:    original-tests/tests/toUpper:      2 / 2
22:    original-tests/tests/arithmeticSeq: 4 / 4
23:    original-tests/tests/geometricSeq:  4 / 4
24:    original-tests/tests/arithmeticSeries: 4 / 4
25:    original-tests/tests/geometricSeries: 4 / 4
26:
27: Git Repo: git@gitlab.doc.ic.ac.uk:lab2324_autumn/haskellsequences_anb122.git
28: Commit ID: b0f37
```

```
1: module Sequences where
2:
3: import Data.Char (ord, chr)
4:
5: -- | Returns the first argument if it is larger than the second,
6: -- the second argument otherwise
7: maxOf2 :: Int -> Int -> Int
8: maxOf2 x y
9:   | x > y = x
10:  | otherwise = y
11:
12: -- | Returns the largest of three Ints
13: maxOf3 :: Int -> Int -> Int -> Int
14: maxOf3 x y z = maxOf2 x (maxOf2 y z)
15:
16: -- | Returns True if the character represents a digit '0'..'9';
17: -- False otherwise
18: isADigit :: Char -> Bool
19: isADigit x = ord x >= 48 && ord x <= 57
20:
21: -- | Returns True if the character represents an alphabetic
22: -- character either in the range 'a'..'z' or in the range 'A'..'Z';
23: -- False otherwise
24: isAlpha :: Char -> Bool
25: isAlpha x
26:   | ord x >= 65 && ord x <= 90 = True
27:   | ord x >= 97 && ord x <= 122 = True
28:   | otherwise = False
29:
30: -- | Returns the integer [0..9] corresponding to the given character.
31: -- Note: this is a simpler version of digitToInt in module Data.Char,
32: -- which does not assume the precondition.
33: digitToInt :: Char -> Int
34: -- Pre: the character is one of '0'..'9'
35: digitToInt x = (ord x) - 48
36:
37: -- | Returns the upper case character corresponding to the input.
38: -- Uses guards by way of variety.
39: toUpper :: Char -> Char
40: toUpper x
41:   | ord x >= 97 && ord x <= 122 = chr ((ord x) - 32)
42:   | otherwise = x
43:
44:
45: --
46: -- Sequences and series
47: --
48:
49: -- | Arithmetic sequence
50: arithmeticSeq :: Double -> Double -> Int -> Double
51: arithmeticSeq a d n = a + ((fromIntegral n) * d)
52:
53: -- | Geometric sequence
54: geometricSeq :: Double -> Double -> Int -> Double
55: geometricSeq a r n = a * (r ** (fromIntegral n))
56:
57: -- | Arithmetic series
58: arithmeticSeries :: Double -> Double -> Int -> Double
59: arithmeticSeries a d n = ((fromIntegral n)+1) * (a + ((d*(fromIntegral n))/2))
60:
61: -- | Geometric series
62: geometricSeries :: Double -> Double -> Int -> Double
63: geometricSeries a r n
64:   | r == 1 = a * ((fromIntegral n) + 1)
65:   | otherwise = a * ((1-(r ** ((fromIntegral n) + 1)))/(1-r))
```

Test Preview**testResults.txt: 1/1****Adithya Narayanan - anb122:j1:24**

```
1: ----- Test Output -----
2: [1 of 3] Compiling IC.TestSuite      ( IC/TestSuite.hs, IC/TestSuite.o )
3: [2 of 3] Compiling Sequences        ( Sequences.hs, Sequences.o )
4: [3 of 3] Compiling Tests            ( Tests.hs, Tests.o )
5: maxOf2: 3 / 3
6:
7: maxOf3: 3 / 3
8:
9: isADigit: 2 / 2
10:
11: isAlpha: 2 / 2
12:
13: digitToInt: 2 / 2
14:
15: toUpper: 2 / 2
16:
17: arithmeticSeq: 4 / 4
18:
19: geometricSeq: 4 / 4
20:
21: arithmeticSeries: 4 / 4
22:
23: geometricSeries: 4 / 4
24:
25: copying Tests.hs from skeleton
26: [3 of 3] Compiling Tests            ( Tests.hs, Tests.o )
27: maxOf2: 3 / 3
28:
29: maxOf3: 3 / 3
30:
31: isADigit: 2 / 2
32:
33: isAlpha: 2 / 2
34:
35: digitToInt: 2 / 2
36:
37: toUpper: 2 / 2
38:
39: arithmeticSeq: 4 / 4
40:
41: geometricSeq: 4 / 4
42:
43: arithmeticSeries: 4 / 4
44:
45: geometricSeries: 4 / 4
46:
47:
48: ----- Test Errors -----
49: Checking https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/
50: Checked https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/
51: Downloading https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/
52: Downloaded https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/
53: Checking https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
54: Checked https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
55: Downloading https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
56: Downloaded https://repol.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
```