



University of Arizona, Department of Computer Science

CSc 372 — Assignment 1 — Due noon, Mon Sep 12 — 5%

Christian Collberg  
September 2, 2005

## 1 Introduction

The purpose of this assignment is to get started writing Haskell functions. For the purposes of this assignment, don't use any of the higher-order built-in functions such as `map`, `foldr`, etc. — I want you to write all functions “from scratch”! You may use the `++`-function for string concatenation.

## 2 Simple Non-Recursive Functions

1. Define a function `doublestring s` which takes a string argument `s` and returns a new string consisting of two copies of `s`: [10 points]

```
doublestring :: String -> String
doublestring s = ...
```

```
> doublestring ""
""
> doublestring "hello"
"hellohello"
```

2. Write a function `charToString a` which returns the one-character string consisting only of the character `a`: [10 points]

```
charToString Char -> String
charToString c = ...
```

```
> charToString 'A'
"A"
```

3. Using the formula

$$V = 2\pi rh + 2\pi r^2$$

define a function `cylinderSurfaceArea (r,h)` which computes the surface area of a cylinder of height `r` and radius `h`: [10 points]

```
cylinderSurfaceArea :: (Float,Float) -> Float
cylinderSurfaceArea (r,h) = ...
```

```
> cylinderSurfaceArea (2.0,5.0)
87.96459420000001
```

4. Write a non-recursive function `yahtzee xs` which takes a list of five `Ints` (between 1 and 6) as argument (the result of rolling five dice) and returns `True` if all numbers are the same, and `False` otherwise. [10 points]

```
yahtzee :: [Int] -> Bool
yahtzee xs = ...
```

```
> yahtzee [1,1,1,1,1]
True
> yahtzee [1,1,1,1,2]
False
> yahtzee [6,6,6,6,6]
True
```

You don't have to check the input for correctness — we assume that there are exactly five elements in the list and that the numbers are between 1 and 6.

5. Write a non-recursive function `third xs` which returns the third element of a list of `Ints`: [10 points]

```
third :: [Int] -> Int
third xs = ...
```

```
> third [1,2,3,4,5]
3
```

You don't have to check that the list contains at least three elements.

### 3 Simple Recursive Functions

1. Write a recursive function `sum n` that returns the sum of the integers  $1 + 2 + 3 + \dots + n$ , where  $n \geq 0$ : [10 points]

```
sum :: Int -> Int
sum n = ...
```

```
> sum 0
0
> sum 1
1
> sum 5
15
```

`sum` should make use of a *conditional expression*.

2. Write a recursive function `gsum n` that returns the sum of the integers  $1 + 2 + 3 + \dots + n$ , where  $n \geq 0$ : [10 points]

```
gsum :: Int -> Int
gsum ...
```

`gsum` should make use of *guards*.

3. Define a recursive function `copystring (s,n)` which returns a string consisting of `n` copies of the string `s`: [10 points]

```
copystring :: (String,Int) -> String
copystring (s,n) = ...
```

Your function should have the following behavior:

```
> copystring ("hello",-1)
""
> copystring ("hello",0)
""
> copystring ("hello", 1)
"hello"
> copystring ("hello",2)
"hellohello"
> copystring ("hello",10)
"hellohellohellohellohellohellohellohellohello"
```

4. Write recursive function `numlist n` which generates a list of the integers  $[0, 1, 2, 3, \dots, n]$ , where  $n \geq 0$ . Generate the error "illegal argument" for  $n < 0$ . [10 points]

```
numlist :: Int -> [Int]
numlist n = ...

> numlist 0
[0]
> numlist 4
[0,1,2,3,4]
> numlist -5
program error: illegal argument
```

5. Write a recursive function `ryahtzee xs` which takes a list of five `Ints` (between 1 and 6) as argument (the result of rolling five dice) and returns `True` if all numbers are the same, and `False` otherwise. [10 points]

```
ryahtzee :: [Int] -> Bool
ryahtzee xs = ...

> ryahtzee [1,1,1,1,1]
True
> ryahtzee [1,1,1,1,2]
False
> ryahtzee [6,6,6,6,6]
True
```

You don't have to check the input for correctness — we assume that there are exactly five elements in the list and that the numbers are between 1 and 6.

## 4 Submission and Assessment

The deadline for this assignment is noon, Mon Sep 12. It is worth 5% of your final grade.

You should submit the assignment electronically using the `Unix` command

```
turnin cs372.1 ass1.hs
```

**Don't show your code to anyone, don't read anyone else's code, don't discuss the details of your code with anyone. If you need help with the assignment see the instructor or the TA.**