Computer Science & Engineering Department
I. I. T. Kharagpur

Principles of Programming Languages: CS40032

Elective

Assignment – 3: Haskell, Scheme and Lisp *Marks: 30*

Solutions

Haskell

1. Func and Func1 are functions which takes multiple arguments. **[1]**

   ```
   ghci> Func1 a b c (Func 2 5)
   ```

   Explain the execution order of the following function. Hint: Curried Function

2. What is the position of (2,2) in **[1]**

   ```
   [ (i,k-i) | k <- [0..],i <- [0..k] ]?
   ```

   Remember that list positions start with 0

   .

3. What is the output of the following **[4]**

   a)
   ```
   ghci> [x*y | x <- [1..5], y <- take 2 (cycle [1,2]) ]
   ```

   b)
   ```
   ghci> let lst = ["cat", "dog", "ant", "pen"]
   ghci> map reverse lst
   ```

   c)
   ```
   ghci> let xxs = [[1,3,5,2,3,1,2,4,5],
                    [4..8],
                    [1,2,4,2,1,6,3,1,3,2,3,6]]
   ghci> [ [ x | x <- xs, odd x ] | xs <- [1,7,4,8,2]:xxs]
   ```

   d)
   ```
   ghci> [3,2,1] > [2,10,200]
   ```

4. Write a function to insert an element in a list at desired index in Haskell using recursion.(Assume the desired index to be within the length of the list.) **[2]**

   ```
   ghci> insertElement 'k' ['a', 'b', 'c', 'd'] 2 => "abkcd"
   ```

5. Write a function extractNonUppercase which extracts the uppercase letters from a string. extractNonUppercase "TYuiJ" would produce "TYJ". Mention the type of the function. **[2]**

Scheme

6. Find the output of the following statements. Explain the intermediate steps. **[4]**

   a)
   ```
   (car (cdr '(a b c d e f)))
   ```

   b)
   ```
   (let ((x 9))
       (* x
           (let ((x (/ x 3)))
               (+ x x))))
   ```

   c)
   ```
   (quote (quote cons))
   ```

   d)
   ```
   (map  (lambda (ls) (cons 'a ls))  '((b c) (a) ()))
   ```

7. Write a Scheme function to compute the sum of the numbers in a list. Return 0, if the list is empty. **[2]**

8. Write Scheme functions using lambda expressions and conditionals to implement: **[4]**

```
A) Counting the number of elements in a list.
   For example (Counter '(a b c b a)) returns 5
B) Logical "AND" operator
```

Lisp

9. Write a LISP program using lambda expression for a function that takes a list and an integer as parameter and returns the list with the elements multiplied by that integer. **[5]**

10. Define a power function (Power x y implies x^y) using recursion constructs of LISP **[5]**