Universidad de Puerto Rico
Recinto Universitario de Mayagüez

ICOM 4036 – Programming Languages

Otoño 2003

Ejercicios de práctica
Examen Parcial I

1. Write an iterative procedure in FORTRAN to compute and return the greatest common divisor (GCD) of 2 integer arguments. First write the procedure in a FORTRAN and then compile the HLL code to MIPS assembly. The GCD can be defined recurrently as:

    a. GCD(a, b) = b if b divides a

    b. GCD(a, b) = GCD(b,r) otherwise, where r = a MOD b

2. Write an recursive procedure in Scheme to compute and return the greatest common divisor (GCD) of 2 integer arguments.

3. Write a FORTRAN iterative procedure called ***precision***() with no arguments. The procedure must return the smallest floating point number e that can be added to 1 such that the result of the sum e+1 is different from 1.

4. Write a tail recursive version in Scheme of the procedure developed in problem 4.

5. Translate the procedure implemented in problem 4 to a tail recursive version using Scheme

6. Write a procedure ***roots(a,b,c)*** that receives the three float coefficients of a polynomial and returns an integer representing the number of distinct real roots.

7. Write FORTRAN procedures sin(x,p), tan(x,p) and exp(x,p) that take two REAL arguments. The procedure should return the approximated floating point value of the function by computing the sum of an appropriate Taylor series. The result should be approximated to a precision p, that is, the the value returned by the function should not differ from the exact value by more than p.

```
(define (fold-left op initial sequence)
     (define (iter result rest)
          (if (null? rest)
               result
               (iter (op result (car rest)) (cdr rest))))
     (iter initial sequence))
```

Describe in your own word what the function fold-left does.

What is the value returned by the following expressions:

a) `(fold-left + 1 (list 1 2 3))`

b) `(fold-left / 1 (list 1 2 3))`

Complete the following definition of a function reverse returns argument list with its elements reversed:

```
(define (reverse sequence)
     (fold-left (lambda (x y) ????? ) nill sequence)
```