Instruction Manual of Auto-Grader (Racket)

All the students should have defined a global variable called \*name\*, and you can give this information about the assignment in class. The code should look like:

(define “student’s name” “my name”).

The instructor will write in the file:

(make-asmt “name of assignment” “name of class” “date” (list of test structs)).

The first three variables will be strings while the test structs should be a list.

The instructor should use the make-test function with inputs: function test name as string, designated points per match, the name of the function as a symbol, a list of lists of inputs, and a solution function that the instructor provides. An example is given below:

(make-test "FACTORIAL FUNCTION"

1 ;; 1 point per match

'facty ;; name of function (a symbol)

'(0 1 2 3 8 10) ;; list of inputs

(letrec ((facty-soln (lambda (n)

(if (<= n 1)

1

(\* n (facty-soln (- n 1)))))))

facty-soln))

You should define which assert function you would like to use to from the assert file to make sure that the student’s answer and the solution are equal. There is already a pre-existing file of asserts that you can chose from, but if there is not an assert that can be used, please define your own assert function.

The auto-grade will fetch contents of TEST data structure and run the student’s solution against yours. If answers do not match, student will be awarded 0 points; if not, they will get however many points you assign.

After you run the program, the statement that displays is:

--------------------------------------------

STUDENT: john doe

ASMT info: items in asmt struct (not including test struct)

------------------------

Test FACTY:

expected output student output points

input(s): 0 1 1 1

input(s): 1 1 2 0

---------------------------------------------------------------------

subtotal 1

(the other tests)

----------------------------------

global score XXX out of possible YYYY

If the student’s function throws an error, the error should be printed out along with the above.