

| 11iM-Λ[M-ΛT 10, 16 10:41 | hw07-LETREC-output | Page 1/3 |
|--------------------------|--|----------|
| 1 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/fact-of-5 | |
| 2 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 3 | > % ===== | |
| 4 | % fact. (value=120) | |
| 5 | | |
| 6 | letrec fact(n) | |
| 7 | = if zero?(n) | |
| 8 | then 1 | |
| 9 | else *(n,(fact -(n, 1))) | |
| 10 | in (fact 5) | |
| 11 | | |
| 12 | % ===== | |
| 13 | 120 | |
| 14 | > | |
| 15 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/letrec-1 | |
| 16 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 17 | > % ===== | |
| 18 | % letrecs. (value=32) | |
| 19 | | |
| 20 | letrec f(x) | |
| 21 | = -(x,1) | |
| 22 | in (f 33) | |
| 23 | | |
| 24 | % ===== | |
| 25 | 32 | |
| 26 | > | |
| 27 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/letrec-2 | |
| 28 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 29 | > % ===== | |
| 30 | % letrecs. (value=8) | |
| 31 | | |
| 32 | letrec f(x) | |
| 33 | = if zero?(x) | |
| 34 | then 0 | |
| 35 | else -((f -(x,1)),-2) | |
| 36 | in (f 4) | |
| 37 | | |
| 38 | % ===== | |
| 39 | 8 | |
| 40 | > | |
| 41 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/letrec-3 | |
| 42 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 43 | > % ===== | |
| 44 | % letrecs. (value=20) | |
| 45 | | |
| 46 | let m=-5 | |
| 47 | in letrec f(x) | |
| 48 | = if zero?(x) | |
| 49 | then 0 | |
| 50 | else -((f -(x,1)),m) | |
| 51 | in (f 4) | |
| 52 | | |
| 53 | % ===== | |
| 54 | 20 | |
| 55 | > | |
| 56 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/letrec-double | |
| 57 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 58 | > % ===== | |
| 59 | % recursive double. (value=12) | |
| 60 | | |
| 61 | letrec double(x) | |
| 62 | = if zero?(x) | |
| 63 | then 0 | |
| 64 | else -(double -(x,1)),-(0, 2)) | |
| 65 | in (double 6) | |
| 66 | | |

| 11iM-Λ[M-ΛT 10, 16 10:41 | hw07-LETREC-output | Page 2/3 |
|--------------------------|--|----------|
| 67 | % ===== | |
| 68 | 12 | |
| 69 | > | |
| 70 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/letrec-sumto | |
| 71 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 72 | > % ===== | |
| 73 | % recursive sumto. (value=55) | |
| 74 | | |
| 75 | letrec sumto(n) | |
| 76 | = if zero?(n) | |
| 77 | then 0 | |
| 78 | else -((sumto -(n,1)),-(0,n)) | |
| 79 | in (sumto 10) | |
| 80 | | |
| 81 | % ===== | |
| 82 | 55 | |
| 83 | > | |
| 84 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/proc-apply-1 | |
| 85 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 86 | > % ===== | |
| 87 | % proc application. (value=29) | |
| 88 | | |
| 89 | (proc(x)-(x,1) 30) | |
| 90 | | |
| 91 | % ===== | |
| 92 | 29 | |
| 93 | > | |
| 94 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/proc-apply-2 | |
| 95 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 96 | > % ===== | |
| 97 | % proc application. (value=29) | |
| 98 | | |
| 99 | let f=proc(x) -(x,1) | |
| 100 | in (f 30) | |
| 101 | | |
| 102 | % ===== | |
| 103 | 29 | |
| 104 | > | |
| 105 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/proc-apply-3 | |
| 106 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 107 | > % ===== | |
| 108 | % proc can be applied twice. p75. (value=55) | |
| 109 | | |
| 110 | let f=proc(x) -(x,11) | |
| 111 | in (f (f 77)) | |
| 112 | | |
| 113 | % ===== | |
| 114 | 55 | |
| 115 | > | |
| 116 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/proc-currying-1 | |
| 117 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 118 | > % ===== | |
| 119 | % multiple arguments with proc that returns proc. (value=-1) | |
| 120 | % (This is called Currying.) | |
| 121 | | |
| 122 | ((proc(x)proc(y)-(x,y) 5) 6) | |
| 123 | | |
| 124 | % ===== | |
| 125 | -1 | |
| 126 | > | |
| 127 | [dshin@acacia letrec]\$ letrec /home/pl/hw07/tests/proc-currying-2 | |
| 128 | Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. | |
| 129 | > % ===== | |
| 130 | % multiple arguments with proc that returns proc. (value=-1) | |
| 131 | % (This is called Currying.) | |
| 132 | | |

11iM-Λ[M-ΛT 10, 16 10:41

hw07-LETREC-output

Page 3/3

```

133 let f=proc(x)proc(y)-(x,y)
134 in ((f 5) 6)
135
136 % =====
137 -1
138 >
139 [dshin@acacia letrec]$ letrec /home/pl/hw07/tests/proc-carrying-3
140 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
141 > % =====
142 % multiple arguments with proc that returns proc. (value=40)
143 % (This is called Currying.)
144
145 let plus=proc(x) proc(y) -(x,-(0,y))
146 in let minus=proc(x) proc(y) -(x,y)
147     in ((minus ((plus 10) 20)) ((minus 40) 50))
148
149 % =====
150 40
151 >
152 [dshin@acacia letrec]$ letrec /home/pl/hw07/tests/proc-higher-1
153 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
154 > % =====
155 % proc's arg is proc. (value=29)
156 % (This is called higher-order function.)
157
158 (proc(f) (f 30)
159  proc(x) -(x,1))
160
161 % =====
162 29
163 >
164 [dshin@acacia letrec]$ letrec /home/pl/hw07/tests/proc-higher-2
165 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
166 > % =====
167 % proc's arg is proc. p75. (value=55)
168 % (This is called higher-order function.)
169
170 (proc(f) (f (f 77))
171  proc(x) -(x,11))
172
173 % =====
174 55
175 >
176 [dshin@acacia letrec]$ letrec /home/pl/hw07/tests/proc-in-let
177 Welcome to MzScheme v370 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.
178 > % =====
179 % procs in let is very useful. p76. (value=-100)
180
181 let x=200
182 in let f=proc(z) -(z,x)
183     in let x=100
184         in let g=proc(z) -(z,x)
185             in -((f 1), (g 1))
186
187 % =====
188 -100
189 >
190 [dshin@acacia letrec]$

```