Quiz on RScript
20170115 20170115

```
Q1. # Double nested function using global y/k3
y <- 6
k3 <- function( ) {
  function( ) y
}
y <- 7
cat("k3( )( )=",k3( )( ),"\n",sep="")
# (A)4. (B)12. (C)8. (D)7. (E)9.
A1. [  ]
Q2. ## Global x returned
x <- 10
n <- function(x){
  x <<- x + 8
}
cat("n(2)=",n(2),’\n’,sep="")
# (A)7. (B)13. (C)10. (D)9. (E)12.
A2. [  ]
Q3. # Inner closure
y <- 4
u <- 6
x <- 7
k1 <- function(y) {
  k2 <- function(y) {
    y <<- 5
    u <<- 8
    x <<- 2
    k3 <- function(u) {
      u * 10
    }
    k3(y) + 6
  }
  k2(y) * x
}
cat("k1(7)=",k1(7),"\n",sep=’’)
# (A)113. (B)114. (C)109. (D)112. (E)110.
A3. [  ]
Q4. # Double nested function using global u/k1
u <- 9
k1 <- function( ) {
  function( ) u
}
u <- 4
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)4. (B)2. (C)-1. (D)3. (E)6.
A4. [  ]
Q5. # Double nested identity function/k3
z <- 0
k3 <- function(z) { # z unused
  function(z) z
}
z <- 1
cat("k3(z)(z)=",k3(z)(z),"\n",sep="")
# (A)1. (B)-1. (C)0. (D)7. (E)4.
A5. [  ]
Q6. # Double nested function using global x/k1
u <- 2
x <- 5
```

```
k1 <- function(u) {
  function( ) {
    u + x # which u is used (global u or k1(u))?
  }
}
u <- 3
x <- 9
cat("k1(0)( )=",k1(0)( ),"\n",sep="")
# (A)13. (B)7. (C)8. (D)9. (E)6.
A6. [  ]
Q7. # Single function, lexical scoping/k3
z <- 7
k3 <- function(z) {
  z + 5
}
z <- 9
cat("k3(z)=",k3(z),"\n",sep="")
# (A)12. (B)18. (C)14. (D)19. (E)13.
A7. [  ]
Q8. # Double nested identity function/k2
y <- 1
k2 <- function(y) { # y unused
  function(y) y
}
y <- 5
cat("k2(y)(y)=",k2(y)(y),"\n",sep="")
# (A)5. (B)4. (C)6. (D)1. (E)7.
A8. [  ]
Q9. # Local z returned
z <- 3
n <- function(z){
  z <- z + 4
}
cat("n(6)=",n(6),’\n’,sep="")
# (A)9. (B)13. (C)7. (D)12. (E)10.
A9. [  ]
Q10. ## Global y returned
y <- 4
n <- function(y){
  y <<- y + 10
}
cat("n(9)=",n(9),’\n’,sep="")
# (A)21. (B)19. (C)20. (D)17. (E)22.
A10. [  ]
Q11. # Double nested function using global x/k3
y <- 5
x <- 6
k3 <- function(y) {
  function( ) {
    y + x # which y is used (global y or k3(y))?
  }
}
y <- 0
x <- 9
cat("k3(2)( )=",k3(2)( ),"\n",sep="")
# (A)7. (B)9. (C)13. (D)11. (E)10.
A11. [  ]
Q12. # Double nested function/n
z <- 5
u <- 3
```

```
y <- 8
n <- function(x) {
  u <- x
  function(v) {
    z + u + v # which z, u are used?
  }
}
z <- 9
u <- 10
y <- 0
cat("n(u)(y)=",n(u)(y),"\n",sep="")
# (A)17. (B)19. (C)23. (D)18. (E)20.
A12. [  ]
Q13. # Local u returned
u <- 1
n <- function(u){
  u <- u + 5
}
cat("n(2)=",n(2),’\n’,sep="")
# (A)7. (B)9. (C)8. (D)10. (E)6.
A13. [  ]
Q14. # Closure substractor
k3 <- function(y=10) {
  function(u) {
    u - y
  }
}
cat("k3(5)(10)=",k3(5)(10),"\n")
# (A)5. (B)7. (C)-2. (D)8. (E)4.
A14. [  ]
Q15. # Closure substractor
k2 <- function(u=7) {
  function(y) {
    y - u
  }
}
cat("k2(4)(7)=",k2(4)(7),"\n")
# (A)-3. (B)6. (C)3. (D)7. (E)-1.
A15. [  ]
Q16. # Single function, lexical scoping/k2
u <- 8
k2 <- function( ) {
  u + 2
}
u <- 0
cat("k2( )=",k2( ),"\n",sep="")
# (A)4. (B)3. (C)0. (D)2. (E)1.
A16. [  ]
Q17. # Inner closure
z <- 3
u <- 6
x <- 4
k1 <- function(z) {
  k3 <- function(z) {
    z <<- 0
    u <<- 8
    x <<- 2
    k2 <- function(u) {
      u * 10
    }
    k2(z) + 6
  }
  k3(z) * x
}
cat("k1(4)=",k1(4),"\n",sep=’’)
# (A)12. (B)14. (C)16. (D)9. (E)11.
A17. [  ]
Q18. # Inner closure
y <- 5
x <- 9
u <- 0
k1 <- function(y) {
  k3 <- function(y) {
    y <<- 1
    x <<- 2
    u <<- 4
    k2 <- function(x) {
      x * 10
    }
    k2(y) + 9
  }
  k3(y) * u
}
cat("k1(0)=",k1(0),"\n",sep=’’)
# (A)74. (B)77. (C)75. (D)76. (E)79.
A18. [  ]
Q19. # Triple nested global k2.
u <- 7
x <- 4
k2 <- function(u) {
  u <- 2
  k1 <- function(z) {
    z <- 6
    k3 <- function(y) {
      y * 2 + x
    }
    k3(z) + 10 + x
  }
  k1(u) * 1 + x
}
cat("k2(4)=",k2(4),"\n",sep="")
# (A)36. (B)32. (C)34. (D)37. (E)33.
A19. [  ]
Q20. # Global modified, local returned.
z <- 4
u <- 10 # unused
k3 <- function(u){
  z <<- u + 5
  z # local z returned
}
cat("k3(9)=",k3(9),’\n’,sep="")
# (A)13. (B)12. (C)14. (D)15. (E)16.
A20. [  ]
Q21. # Triple nested global k2.
z <- 4
x <- 2
k2 <- function(z) {
  z <- 2
  k1 <- function(y) {
    y <- 1
    k3 <- function(u) {
      u * 2 + x
    }
    k3(y) + 5 + x
  }
  k1(z) * 6 + x
}
cat("k2(2)=",k2(2),"\n",sep="")
# (A)69. (B)71. (C)62. (D)68. (E)66.
A21. [  ]
Q22. # Double nested function/n
y <- 3
x <- 1
z <- 7
n <- function(u) {
  x <- u
  function(v) {
    y + x + v # which y, x are used?
  }
}
y <- 2
x <- 8
z <- 5
cat("n(x)(z)=",n(x)(z),"\n",sep="")
# (A)13. (B)14. (C)16. (D)20. (E)15.
A22. [  ]
Q23. # Double nested identity function/k1
z <- 1
k1 <- function(z) { # z unused
  function(z) z
}
z <- 8
cat("k1(z)(z)=",k1(z)(z),"\n",sep="")
# (A)5. (B)6. (C)13. (D)8. (E)7.
A23. [  ]
Q24. # Triple nested function/k2
k2 <- function(x) {
  k1 <- function(x) {
    k3 <- function(x) {
      x + 6
    }
    k3(x) + 4
  }
  k1(x) + 4
}
cat("k2(8)=",k2(8),"\n",sep="")
# (A)24. (B)25. (C)17. (D)21. (E)22.
A24. [  ]
Q25. # Single function, lexical scoping/k3
x <- 7
k3 <- function(x) {
  x + 5
}
x <- 4
cat("k3(x)=",k3(x),"\n",sep="")
# (A)9. (B)5. (C)8. (D)12. (E)11.
A25. [  ]
Q26. # Single function, lexical scoping/k1
u <- 7
k1 <- function( ) {
  u + 2
}
u <- 1
cat("k1( )=",k1( ),"\n",sep="")
# (A)4. (B)-1. (C)6. (D)8. (E)3.
A26. [  ]
Q27. # Single function, lexical scoping/k1
y <- 10
k1 <- function(y) {
  y + 9
}
y <- 2
cat("k1(y)=",k1(y),"\n",sep="")
# (A)9. (B)17. (C)11. (D)12. (E)10.
A27. [  ]
Q28. # Two functions, lexical scoping
a <- 1
k1 <- function(z) {
  a * z
}
a <- 10
k2 <- function(u) {
  a <- 100
  k1(u)
}
cat("k2(3)=",k2(3),"\n",sep="")
# (A)33. (B)30. (C)31. (D)27. (E)29.
A28. [  ]
Q29. # Double nested function using global y/k3
x <- 10
y <- 5
k3 <- function(x) {
  function( ) {
    x + y # which x is used (global x or k3(x))?
  }
}
x <- 7
y <- 6
cat("k3(9)( )=",k3(9)( ),"\n",sep="")
# (A)17. (B)15. (C)11. (D)14. (E)10.
A29. [  ]
Q30. # Triple nested global k3.
x <- 0
z <- 2
k3 <- function(x) {
  x <- 2
  k1 <- function(y) {
    y <- 4
    k2 <- function(u) {
      u * 2 + z
    }
    k2(y) + 3 + z
  }
  k1(x) * 7 + z
}
cat("k3(2)=",k3(2),"\n",sep="")
# (A)106. (B)108. (C)107. (D)105. (E)110.
A30. [  ]
Q31. # Two functions, lexical scoping
a <- 1
```

```
k3 <- function(u) {
  a * u
}
a <- 10
k2 <- function(x) {
  a <- 100
  k3(x)
}
cat("k2(3)=",k2(3),"\n",sep="")
# (A)28. (B)26. (C)34. (D)31. (E)30.
A31. [  ]
Q32. # Nested functions, lexical scoping
k1 <- function(u) {
  k3 <- function(z) {
    return(u * z)
  }
  return(k3)
}
cat("k1(7)(9)=",k1(7)(9),"\n",sep="")
# (A)65. (B)61. (C)59. (D)63. (E)62.
A32. [  ]
Q33. # Triple nested function/k3
k3 <- function(x) {
  k2 <- function(x) {
    k1 <- function(x) {
      x + 3
    }
    k1(x) + 6
  }
  k2(x) + 6
}
cat("k3(2)=",k3(2),"\n",sep="")
# (A)17. (B)21. (C)18. (D)13. (E)14.
A33. [  ]
Q34. # Triple nested global k2.
u <- 2
y <- 9
k2 <- function(u) {
  u <- 2
  k3 <- function(x) {
    x <- 10
    k1 <- function(z) {
      z * 2 + y
    }
    k1(x) + 8 + y
  }
  k3(u) * 0 + y
}
cat("k2(9)=",k2(9),"\n",sep="")
# (A)13. (B)10. (C)9. (D)11. (E)7.
A34. [  ]
Q35. # Two functions, lexical scoping
a <- 1
k2 <- function(y) {
  a * y
}
a <- 10
k1 <- function(u) {
  a <- 100
  k2(u)
}
cat("k1(3)=",k1(3),"\n",sep="")
# (A)29. (B)31. (C)35. (D)30. (E)28.
A35. [  ]
Q36. # Double nested function using global u/k1
z <- 0
u <- 10
k1 <- function(z) {
  function( ) {
    z + u # which z is used (global z or k1(z))?
  }
}
z <- 2
u <- 1
cat("k1(3)( )=",k1(3)( ),"\n",sep="")
# (A)3. (B)4. (C)6. (D)1. (E)5.
A36. [  ]
Q37. # Double nested function/n
y <- 7
z <- 0
u <- 6
n <- function(x) {
  z <- x
  function(v) {
    y + z + v # which y, z are used?
  }
}
y <- 8
z <- 2
u <- 1
cat("n(z)(u)=",n(z)(u),"\n",sep="")
# (A)10. (B)9. (C)11. (D)17. (E)12.
A37. [  ]
Q38. # Closure substractor
k3 <- function(x=3) {
  function(y) {
    y - x
  }
}
cat("k3(0)(3)=",k3(0)(3),"\n")
# (A)-5. (B)-4. (C)-6. (D)3. (E)7.
A38. [  ]
Q39. # Local u returned
u <- 4
n <- function(u){
  u <- u + 5
}
cat("n(6)=",n(6),’\n’,sep="")
# (A)13. (B)12. (C)10. (D)9. (E)11.
A39. [  ]
Q40. # Closure substractor
k3 <- function(z=1) {
  function(x) {
    x - z
  }
}
cat("k3(9)(1)=",k3(9)(1),"\n")
# (A)-5. (B)-10. (C)-8. (D)8. (E)4.
A40. [  ]
Q41. # Triple nested global k2.
```

```
z <- 0
y <- 10
k2 <- function(z) {
  z <- 2
  k3 <- function(x) {
    x <- 2
    k1 <- function(u) {
      u * 2 + y
    }
    k1(x) + 3 + y
  }
  k3(z) * 1 + y
}
cat("k2(10)=",k2(10),"\n",sep="")
# (A)34. (B)37. (C)33. (D)38. (E)36.
A41. [  ]
Q42. # Single function, lexical scoping/k2
x <- 2
k2 <- function( ) {
  x + 4
}
x <- 1
cat("k2( )=",k2( ),"\n",sep="")
# (A)5. (B)3. (C)1. (D)4. (E)6.
A42. [  ]
Q43. # Global modified, local returned.
u <- 5
y <- 0 # unused
k1 <- function(y){
  u <<- y + 4
  u # local u returned
}
cat("k1(7)=",k1(7),’\n’,sep="")
# (A)15. (B)12. (C)8. (D)11. (E)9.
A43. [  ]
Q44. # Single function, lexical scoping/k2
z <- 1
k2 <- function(z) {
  z + 6
}
z <- 9
cat("k2(z)=",k2(z),"\n",sep="")
# (A)15. (B)18. (C)13. (D)21. (E)16.
A44. [  ]
Q45. # Single function, lexical scoping/k1
y <- 0
k1 <- function(y) {
  y + 7
}
y <- 9
cat("k1(y)=",k1(y),"\n",sep="")
# (A)17. (B)13. (C)16. (D)15. (E)11.
A45. [  ]
Q46. # Double nested function/n
y <- 9
z <- 10
u <- 7
n <- function(x) {
  z <- x
  function(v) {
    y + z + v # which y, z are used?
  }
}
y <- 6
z <- 4
u <- 5
cat("n(z)(u)=",n(z)(u),"\n",sep="")
# (A)16. (B)15. (C)10. (D)11. (E)13.
A46. [  ]
Q47. # Two functions, lexical scoping
a <- 1
k2 <- function(u) {
  a * u
}
a <- 10
k1 <- function(y) {
  a <- 100
  k2(y)
}
cat("k1(3)=",k1(3),"\n",sep="")
# (A)30. (B)28. (C)27. (D)29. (E)34.
A47. [  ]
Q48. # Triple nested function/k1
k1 <- function(x) {
  k2 <- function(x) {
    k3 <- function(x) {
      x + 2
    }
    k3(x) + 0
  }
  k2(x) + 0
}
cat("k1(3)=",k1(3),"\n",sep="")
# (A)4. (B)3. (C)5. (D)7. (E)1.
A48. [  ]
Q49. # Local y returned
y <- 4
n <- function(y){
  y <- y + 3
}
cat("n(1)=",n(1),’\n’,sep="")
# (A)4. (B)0. (C)3. (D)6. (E)5.
A49. [  ]
Q50. # Triple nested function/k1
k1 <- function(u) {
  k2 <- function(u) {
    k3 <- function(u) {
      u + 10
    }
    k3(u) + 1
  }
  k2(u) + 1
}
cat("k1(0)=",k1(0),"\n",sep="")
# (A)11. (B)10. (C)7. (D)8. (E)12.
A50. [  ]
Q51. # Triple nested global k3.
u <- 10
z <- 3
k3 <- function(u) {
  u <- 2
  k1 <- function(y) {
    y <- 7
    k2 <- function(x) {
      x * 2 + z
    }
    k2(y) + 9 + z
  }
  k1(u) * 8 + z
}
cat("k3(3)=",k3(3),"\n",sep="")
# (A)230. (B)235. (C)238. (D)233. (E)232.
A51. [  ]
Q52. # Closure substractor
k1 <- function(y=1) {
  function(u) {
    u - y
  }
}
cat("k1(6)(1)=",k1(6)(1),"\n")
# (A)4. (B)-5. (C)-2. (D)-4. (E)-10.
A52. [  ]
Q53. # Nested functions, lexical scoping
k2 <- function(u) {
  k3 <- function(y) {
    return(u * y)
  }
  return(k3)
}
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)62. (B)64. (C)63. (D)67. (E)65.
A53. [  ]
Q54. # Double nested function using global z/k3
y <- 0
z <- 5
k3 <- function(y) {
  function( ) {
    y + z # which y is used (global y or k3(y))?
  }
}
y <- 6
z <- 3
cat("k3(1)( )=",k3(1)( ),"\n",sep="")
# (A)3. (B)5. (C)4. (D)0. (E)7.
A54. [  ]
Q55. ## Global y returned
y <- 0
n <- function(y){
  y <<- y + 9
}
cat("n(2)=",n(2),’\n’,sep="")
# (A)11. (B)13. (C)14. (D)9. (E)8.
A55. [  ]
Q56. # Triple nested global k3.
z <- 0
u <- 1
k3 <- function(z) {
  z <- 2
  k1 <- function(y) {
    y <- 10
    k2 <- function(x) {
      x * 2 + u
    }
    k2(y) + 9 + u
  }
  k1(z) * 5 + u
}
cat("k3(1)=",k3(1),"\n",sep="")
# (A)158. (B)156. (C)154. (D)150. (E)155.
A56. [  ]
Q57. # Double nested function using global y/k3
y <- 3
k3 <- function( ) {
  function( ) y
}
y <- 8
cat("k3( )( )=",k3( )( ),"\n",sep="")
# (A)11. (B)9. (C)7. (D)8. (E)5.
A57. [  ]
Q58. # Global modified, local returned.
x <- 2
y <- 4 # unused
k1 <- function(y){
  x <<- y + 8
  x # local x returned
}
cat("k1(10)=",k1(10),’\n’,sep="")
# (A)20. (B)16. (C)21. (D)13. (E)18.
A58. [  ]
Q59. # Double nested function using global x/k3
x <- 7
k3 <- function( ) {
  function( ) x
}
x <- 5
cat("k3( )( )=",k3( )( ),"\n",sep="")
# (A)11. (B)5. (C)4. (D)3. (E)7.
A59. [  ]
Q60. ## Global u returned
u <- 1
n <- function(u){
  u <<- u + 7
}
cat("n(5)=",n(5),’\n’,sep="")
# (A)14. (B)9. (C)12. (D)11. (E)7.
A60. [  ]
Q61. # Double nested function using global u/k2
x <- 2
u <- 4
k2 <- function(x) {
  function( ) {
    x + u # which x is used (global x or k2(x))?
  }
}
x <- 0
u <- 3
cat("k2(9)( )=",k2(9)( ),"\n",sep="")
# (A)7. (B)10. (C)11. (D)12. (E)15.
A61. [  ]
Q62. # Triple nested global k2.
```

```
y <- 8
x <- 1
k2 <- function(y) {
  y <- 2
  k3 <- function(z) {
    z <- 9
    k1 <- function(u) {
      u * 2 + x
    }
    k1(z) + 3 + x
  }
  k3(y) * 0 + x
}
cat("k2(1)=",k2(1),"\n",sep="")
# (A)0. (B)1. (C)6. (D)7. (E)-1.
A62. [  ]
Q63. # Triple nested function/k3
k3 <- function(z) {
  k1 <- function(z) {
    k2 <- function(z) {
      z + 3
    }
    k2(z) + 0
  }
  k1(z) + 0
}
cat("k3(10)=",k3(10),"\n",sep="")
# (A)15. (B)12. (C)19. (D)14. (E)13.
A63. [  ]
Q64. # Triple nested function/k3
k3 <- function(u) {
  k1 <- function(u) {
    k2 <- function(u) {
      u + 3
    }
    k2(u) + 7
  }
  k1(u) + 7
}
cat("k3(9)=",k3(9),"\n",sep="")
# (A)28. (B)22. (C)25. (D)29. (E)26.
A64. [  ]
Q65. # Closure substractor
k1 <- function(u=10) {
  function(y) {
    y - u
  }
}
cat("k1(3)(10)=",k1(3)(10),"\n")
# (A)6. (B)7. (C)-5. (D)0. (E)5.
A65. [  ]
Q66. # Single function, lexical scoping/k2
u <- 2
k2 <- function(u) {
  u + 9
}
u <- 4
cat("k2(u)=",k2(u),"\n",sep="")
# (A)13. (B)16. (C)9. (D)8. (E)15.
A66. [  ]
```

```
Q67. # Local x returned
x <- 2
n <- function(x){
  x <- x + 8
}
cat("n(1)=",n(1),’\n’,sep="")
# (A)11. (B)9. (C)14. (D)7. (E)10.
A67. [  ]
Q68. # Two functions, lexical scoping
a <- 1
k1 <- function(u) {
  a * u
}
a <- 10
k3 <- function(y) {
  a <- 100
  k1(y)
}
cat("k3(3)=",k3(3),"\n",sep="")
# (A)27. (B)30. (C)31. (D)26. (E)28.
A68. [  ]
Q69. # Triple nested function/k1
k1 <- function(y) {
  k3 <- function(y) {
    k2 <- function(y) {
      y + 8
    }
    k2(y) + 3
  }
  k3(y) + 3
}
cat("k1(0)=",k1(0),"\n",sep="")
# (A)11. (B)17. (C)14. (D)12. (E)15.
A69. [  ]
Q70. # Local y returned
y <- 2
n <- function(y){
  y <- y + 7
}
cat("n(10)=",n(10),’\n’,sep="")
# (A)21. (B)17. (C)18. (D)15. (E)20.
A70. [  ]
Q71. # Double nested function/n
x <- 3
z <- 4
y <- 2
n <- function(u) {
  z <- u
  function(v) {
    x + z + v # which x, z are used?
  }
}
x <- 1
z <- 9
y <- 8
cat("n(z)(y)=",n(z)(y),"\n",sep="")
# (A)15. (B)13. (C)23. (D)18. (E)17.
A71. [  ]
Q72. # Single function, lexical scoping/k1
x <- 7
```

```
k1 <- function(x) {
  x + 6
}
x <- 0
cat("k1(x)=",k1(x),"\n",sep="")
# (A)5. (B)7. (C)3. (D)6. (E)8.
A72. [  ]
Q73. # Triple nested global k1.
y <- 4
x <- 6
k1 <- function(y) {
  y <- 2
  k3 <- function(z) {
    z <- 9
    k2 <- function(u) {
      u * 2 + x
    }
    k2(z) + 3 + x
  }
  k3(y) * 8 + x
}
cat("k1(6)=",k1(6),"\n",sep="")
# (A)271. (B)270. (C)273. (D)274. (E)277.
A73. [  ]
Q74. # Single function, lexical scoping/k3
u <- 6
k3 <- function(u) {
  u + 3
}
u <- 8
cat("k3(u)=",k3(u),"\n",sep="")
# (A)10. (B)15. (C)13. (D)14. (E)11.
A74. [  ]
Q75. # Single function, lexical scoping/k3
z <- 7
k3 <- function(z) {
  z + 5
}
z <- 0
cat("k3(z)=",k3(z),"\n",sep="")
# (A)5. (B)8. (C)4. (D)3. (E)1.
A75. [  ]
Q76. # Local y returned
y <- 5
n <- function(y){
  y <- y + 10
}
cat("n(2)=",n(2),’\n’,sep="")
# (A)10. (B)12. (C)8. (D)13. (E)11.
A76. [  ]
Q77. # Triple nested function/k2
k2 <- function(y) {
  k3 <- function(y) {
    k1 <- function(y) {
      y + 7
    }
    k1(y) + 4
  }
  k3(y) + 4
}
```

```
cat("k2(0)=",k2(0),"\n",sep="")
# (A)16. (B)11. (C)15. (D)17. (E)19.
A77. [  ]
Q78. # Double nested function using global u/k1
u <- 9
k1 <- function( ) {
  function( ) u
}
u <- 7
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)5. (B)7. (C)6. (D)8. (E)9.
A78. [  ]
Q79. # Double nested function/n
u <- 8
y <- 9
z <- 3
n <- function(x) {
  y <- x
  function(v) {
    u + y + v # which u, y are used?
  }
}
u <- 10
y <- 6
z <- 4
cat("n(y)(z)=",n(y)(z),"\n",sep="")
# (A)21. (B)22. (C)18. (D)17. (E)20.
A79. [  ]
Q80. # Single function, lexical scoping/k2
z <- 5
k2 <- function( ) {
  z + 7
}
z <- 3
cat("k2( )=",k2( ),"\n",sep="")
# (A)8. (B)10. (C)4. (D)14. (E)11.
A80. [  ]
Q81. ## Global y returned
y <- 10
n <- function(y){
  y <<- y + 0
}
cat("n(4)=",n(4),’\n’,sep="")
# (A)0. (B)-2. (C)2. (D)3. (E)4.
A81. [  ]
Q82. # Local u returned
u <- 8
n <- function(u){
  u <- u + 9
}
cat("n(4)=",n(4),’\n’,sep="")
# (A)12. (B)11. (C)13. (D)17. (E)15.
A82. [  ]
Q83. # Double nested function/n
u <- 6
x <- 3
z <- 7
n <- function(y) {
  x <- y
  function(v) {
    u + x + v # which u, x are used?
  }
}
u <- 1
x <- 9
z <- 10
cat("n(x)(z)=",n(x)(z),"\n",sep="")
# (A)18. (B)24. (C)20. (D)21. (E)19.
A83. [  ]
Q84. # Inner closure
y <- 2
u <- 9
x <- 7
k1 <- function(y) {
  k3 <- function(y) {
    y <<- 10
    u <<- 6
    x <<- 8
    k2 <- function(u) {
      u * 10
    }
    k2(y) + 9
  }
  k3(y) * x
}
cat("k1(7)=",k1(7),"\n",sep=’’)
# (A)873. (B)870. (C)879. (D)872. (E)874.
A84. [  ]
Q85. # Nested functions, lexical scoping
k3 <- function(y) {
  k1 <- function(u) {
    return(y * u)
  }
  return(k1)
}
cat("k3(7)(9)=",k3(7)(9),"\n",sep="")
# (A)59. (B)63. (C)61. (D)64. (E)65.
A85. [  ]
Q86. # Double nested function using global z/k3
x <- 6
z <- 9
k3 <- function(x) {
  function( ) {
    x + z # which x is used (global x or k3(x))?
  }
}
x <- 0
z <- 8
cat("k3(1)( )=",k3(1)( ),"\n",sep="")
# (A)10. (B)11. (C)9. (D)8. (E)4.
A86. [  ]
Q87. # Double nested function using global y/k1
u <- 6
y <- 8
k1 <- function(u) {
  function( ) {
    u + y # which u is used (global u or k1(u))?
  }
}
u <- 10
```

```
y <- 4
cat("k1(9)( )=",k1(9)( ),"\n",sep="")
# (A)13. (B)9. (C)17. (D)12. (E)7.
A87. [  ]
Q88. # Double nested function using global u/k1
x <- 10
u <- 9
k1 <- function(x) {
  function( ) {
    x + u # which x is used (global x or k1(x))?
  }
}
x <- 0
u <- 6
cat("k1(3)( )=",k1(3)( ),"\n",sep="")
# (A)8. (B)9. (C)12. (D)5. (E)10.
A88. [  ]
Q89. # Double nested function using global u/k3
u <- 7
k3 <- function( ) {
  function( ) u
}
u <- 0
cat("k3( )( )=",k3( )( ),"\n",sep="")
# (A)1. (B)-1. (C)3. (D)4. (E)0.
A89. [  ]
Q90. # Inner closure
z <- 4
x <- 2
y <- 8
k2 <- function(z) {
  k3 <- function(z) {
    z <<- 1
    x <<- 6
    y <<- 5
    k1 <- function(x) {
      x * 10
    }
    k1(z) + 2
  }
  k3(z) * y
}
cat("k2(8)=",k2(8),"\n",sep=’’)
# (A)61. (B)57. (C)63. (D)60. (E)58.
A90. [  ]
Q91. # Inner closure
x <- 2
y <- 1
u <- 0
k1 <- function(x) {
  k3 <- function(x) {
    x <<- 3
    y <<- 8
    u <<- 7
    k2 <- function(y) {
      y * 10
    }
    k2(x) + 1
  }
  k3(x) * u
}
cat("k1(0)=",k1(0),"\n",sep=’’)
# (A)217. (B)220. (C)222. (D)221. (E)218.
A91. [  ]
Q92. # Single function, lexical scoping/k2
x <- 2
k2 <- function( ) {
  x + 8
}
x <- 5
cat("k2( )=",k2( ),"\n",sep="")
# (A)11. (B)9. (C)14. (D)13. (E)10.
A92. [  ]
Q93. # Double nested function/n
x <- 0
y <- 7
z <- 4
n <- function(u) {
  y <- u
  function(v) {
    x + y + v # which x, y are used?
  }
}
x <- 9
y <- 6
z <- 5
cat("n(y)(z)=",n(y)(z),"\n",sep="")
# (A)18. (B)20. (C)22. (D)23. (E)19.
A93. [  ]
Q94. # Double nested function using global y/k1
z <- 3
y <- 6
k1 <- function(z) {
  function( ) {
    z + y # which z is used (global z or k1(z))?
  }
}
z <- 10
y <- 5
cat("k1(1)( )=",k1(1)( ),"\n",sep="")
# (A)9. (B)3. (C)7. (D)4. (E)6.
A94. [  ]
Q95. # Double nested identity function/k1
x <- 10
k1 <- function(x) { # x unused
  function(x) x
}
x <- 4
cat("k1(x)(x)=",k1(x)(x),"\n",sep="")
# (A)6. (B)3. (C)4. (D)2. (E)5.
A95. [  ]
Q96. # Double nested identity function/k3
z <- 3
k3 <- function(z) { # z unused
  function(z) z
}
z <- 9
cat("k3(z)(z)=",k3(z)(z),"\n",sep="")
# (A)11. (B)9. (C)10. (D)8. (E)4.
A96. [  ]
```

```
Q97. # Double nested function/n
z <- 4
y <- 10
x <- 7
n <- function(u) {
  y <- u
  function(v) {
    z + y + v # which z, y are used?
  }
}
z <- 3
y <- 6
x <- 0
cat("n(y)(x)=",n(y)(x),"\n",sep="")
# (A)7. (B)11. (C)8. (D)9. (E)6.
A97. [  ]
Q98. # Global modified, local returned.
x <- 9
z <- 8 # unused
k2 <- function(z){
  x <<- z + 10
  x # local x returned
}
cat("k2(7)=",k2(7),’\n’,sep="")
# (A)16. (B)21. (C)13. (D)17. (E)18.
A98. [  ]
Q99. # Single function, lexical scoping/k1
u <- 9
k1 <- function(u) {
  u + 5
}
u <- 10
cat("k1(u)=",k1(u),"\n",sep="")
# (A)11. (B)17. (C)15. (D)16. (E)13.
A99. [  ]
Q100. # Global modified, local returned.
z <- 6
x <- 3 # unused
k3 <- function(x){
  z <<- x + 8
  z # local z returned
}
cat("k3(9)=",k3(9),’\n’,sep="")
# (A)16. (B)19. (C)22. (D)17. (E)18.
A100. [  ]
Q101. # Double nested function using global y/k2
y <- 0
k2 <- function( ) {
  function( ) y
}
y <- 3
cat("k2( )( )=",k2( )( ),"\n",sep="")
# (A)4. (B)3. (C)2. (D)-1. (E)0.
A101. [  ]
Q102. # Double nested identity function/k3
y <- 6
k3 <- function(y) { # y unused
  function(y) y
}
y <- 2
```

```
cat("k3(y)(y)=",k3(y)(y),"\n",sep="")
# (A)6. (B)2. (C)4. (D)3. (E)1.
A102. [  ]
Q103. # Triple nested global k3.
u <- 4
x <- 8
k3 <- function(u) {
  u <- 2
  k1 <- function(y) {
    y <- 9
    k2 <- function(z) {
      z * 2 + x
    }
    k2(y) + 5 + x
  }
  k1(u) * 2 + x
}
cat("k3(8)=",k3(8),"\n",sep="")
# (A)85. (B)86. (C)88. (D)91. (E)87.
A103. [  ]
Q104. # Single function, lexical scoping/k3
u <- 10
k3 <- function( ) {
  u + 4
}
u <- 1
cat("k3( )=",k3( ),"\n",sep="")
# (A)7. (B)5. (C)10. (D)0. (E)6.
A104. [  ]
Q105. # Triple nested function/k1
k1 <- function(z) {
  k2 <- function(z) {
    k3 <- function(z) {
      z + 4
    }
    k3(z) + 1
  }
  k2(z) + 1
}
cat("k1(5)=",k1(5),"\n",sep="")
# (A)7. (B)9. (C)8. (D)11. (E)12.
A105. [  ]
Q106. # Double nested function using global y/k3
z <- 10
y <- 5
k3 <- function(z) {
  function( ) {
    z + y # which z is used (global z or k3(z))?
  }
}
z <- 8
y <- 9
cat("k3(3)( )=",k3(3)( ),"\n",sep="")
# (A)11. (B)13. (C)10. (D)12. (E)14.
A106. [  ]
Q107. # Local x returned
x <- 6
n <- function(x){
  x <- x + 3
}
```

```
cat("n(10)=",n(10),’\n’,sep="")
# (A)9. (B)15. (C)13. (D)7. (E)14.
A107. [  ]
Q108. # Global modified, local returned.
y <- 9
x <- 7 # unused
k3 <- function(x){
  y <<- x + 10
  y # local y returned
}
cat("k3(6)=",k3(6),’\n’,sep="")
# (A)10. (B)12. (C)16. (D)8. (E)14.
A108. [  ]
Q109. # Nested functions, lexical scoping
k3 <- function(u) {
  k1 <- function(x) {
    return(u * x)
  }
  return(k1)
}
cat("k3(7)(9)=",k3(7)(9),"\n",sep="")
# (A)66. (B)65. (C)58. (D)67. (E)63.
A109. [  ]
Q110. # Nested functions, lexical scoping
k3 <- function(z) {
  k1 <- function(y) {
    return(z * y)
  }
  return(k1)
}
cat("k3(7)(9)=",k3(7)(9),"\n",sep="")
# (A)58. (B)60. (C)57. (D)63. (E)62.
A110. [  ]
Q111. # Single function, lexical scoping/k2
u <- 1
k2 <- function( ) {
  u + 5
}
u <- 0
cat("k2( )=",k2( ),"\n",sep="")
# (A)6. (B)2. (C)7. (D)8. (E)5.
A111. [  ]
Q112. # Double nested function using global y/k1
y <- 2
k1 <- function( ) {
  function( ) y
}
y <- 3
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)1. (B)-4. (C)9. (D)2. (E)3.
A112. [  ]
Q113. # Double nested identity function/k3
z <- 4
k3 <- function(z) { # z unused
  function(z) z
}
z <- 2
cat("k3(z)(z)=",k3(z)(z),"\n",sep="")
# (A)4. (B)3. (C)-1. (D)2. (E)0.
A113. [  ]
```

```
Q114. # Nested functions, lexical scoping
k1 <- function(y) {
  k3 <- function(x) {
    return(y * x)
  }
  return(k3)
}
cat("k1(7)(9)=",k1(7)(9),"\n",sep="")
# (A)67. (B)62. (C)61. (D)65. (E)63.
A114. [  ]
Q115. # Double nested identity function/k2
y <- 0
k2 <- function(y) { # y unused
  function(y) y
}
y <- 3
cat("k2(y)(y)=",k2(y)(y),"\n",sep="")
# (A)3. (B)12. (C)6. (D)0. (E)2.
A115. [  ]
Q116. # Closure substractor
k3 <- function(x=9) {
  function(u) {
    u - x
  }
}
cat("k3(0)(9)=",k3(0)(9),"\n")
# (A)6. (B)7. (C)9. (D)-7. (E)1.
A116. [  ]
Q117. # Single function, lexical scoping/k1
x <- 7
k1 <- function(x) {
  x + 2
}
x <- 6
cat("k1(x)=",k1(x),"\n",sep="")
# (A)11. (B)4. (C)9. (D)8. (E)7.
A117. [  ]
Q118. # Double nested identity function/k1
u <- 10
k1 <- function(u) { # u unused
  function(u) u
}
u <- 9
cat("k1(u)(u)=",k1(u)(u),"\n",sep="")
# (A)8. (B)9. (C)10. (D)14. (E)6.
A118. [  ]
Q119. # Inner closure
x <- 8
u <- 10
y <- 3
k3 <- function(x) {
  k2 <- function(x) {
    x <<- 7
    u <<- 2
    y <<- 6
    k1 <- function(u) {
      u * 10
    }
    k1(x) + 10
  }
  k2(x) * y
}
cat("k3(3)=",k3(3),"\n",sep=’’)
# (A)478. (B)480. (C)483. (D)482. (E)485.
A119. [  ]
Q120. # Double nested function using global z/k1
z <- 2
k1 <- function( ) {
  function( ) z
}
z <- 1
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)-2. (B)1. (C)2. (D)3. (E)0.
A120. [  ]
Q121. # Single function, lexical scoping/k1
u <- 10
k1 <- function( ) {
  u + 8
}
u <- 9
cat("k1( )=",k1( ),"\n",sep="")
# (A)16. (B)20. (C)18. (D)17. (E)15.
A121. [  ]
Q122. # Closure substractor
k2 <- function(y=9) {
  function(u) {
    u - y
  }
}
cat("k2(4)(9)=",k2(4)(9),"\n")
# (A)3. (B)5. (C)-7. (D)0.
A122. [  ]
Q123. # Double nested identity function/k1
x <- 10
k1 <- function(x) { # x unused
  function(x) x
}
x <- 7
cat("k1(x)(x)=",k1(x)(x),"\n",sep="")
# (A)6. (B)10. (C)3. (D)8. (E)7.
A123. [  ]
Q124. # Two functions, lexical scoping
a <- 1
k1 <- function(z) {
  a * z
}
a <- 10
k2 <- function(y) {
  a <- 100
  k1(y)
}
cat("k2(3)=",k2(3),"\n",sep="")
# (A)28. (B)32. (C)31. (D)25. (E)30.
A124. [  ]
Q125. # Global modified, local returned.
u <- 6
y <- 3 # unused
k3 <- function(y){
  u <<- y + 4
  u # local u returned
```

```
}
cat("k3(9)=",k3(9),’\n’,sep="")
# (A)13. (B)17. (C)11. (D)15. (E)12.
A125. [  ]
Q126. # Double nested function/n
y <- 8
x <- 1
z <- 0
n <- function(u) {
  x <- u
  function(v) {
    y + x + v # which y, x are used?
  }
}
y <- 2
x <- 3
z <- 6
cat("n(x)(z)=",n(x)(z),"\n",sep="")
# (A)14. (B)11. (C)10. (D)8. (E)12.
A126. [  ]
Q127. ## Global z returned
z <- 9
n <- function(z){
  z <<- z + 10
}
cat("n(8)=",n(8),’\n’,sep="")
# (A)18. (B)20. (C)19. (D)21. (E)16.
A127. [  ]
Q128. # Double nested function/n
u <- 5
y <- 2
x <- 9
n <- function(z) {
  y <- z
  function(v) {
    u + y + v # which u, y are used?
  }
}
u <- 10
y <- 6
x <- 4
cat("n(y)(x)=",n(y)(x),"\n",sep="")
# (A)21. (B)16. (C)18. (D)20. (E)19.
A128. [  ]
Q129. # Single function, lexical scoping/k2
u <- 0
k2 <- function( ) {
  u + 9
}
u <- 3
cat("k2( )=",k2( ),"\n",sep="")
# (A)12. (B)10. (C)13. (D)11. (E)8.
A129. [  ]
Q130. # Triple nested function/k2
k2 <- function(x) {
  k3 <- function(x) {
    k1 <- function(x) {
      x + 10
    }
    k1(x) + 0
  }
  k3(x) + 0
}
cat("k2(4)=",k2(4),"\n",sep="")
# (A)16. (B)12. (C)10. (D)14. (E)15.
A130. [  ]
Q131. # Closure substractor
k1 <- function(u=3) {
  function(y) {
    y - u
  }
}
cat("k1(0)(3)=",k1(0)(3),"\n")
# (A)-5. (B)0. (C)2. (D)1. (E)3.
A131. [  ]
Q132. # Double nested function using global x/k3
y <- 8
x <- 7
k3 <- function(y) {
  function( ) {
    y + x # which y is used (global y or k3(y))?
  }
}
y <- 6
x <- 10
cat("k3(3)( )=",k3(3)( ),"\n",sep="")
# (A)12. (B)13. (C)14. (D)11. (E)15.
A132. [  ]
Q133. # Single function, lexical scoping/k3
y <- 9
k3 <- function( ) {
  y + 8
}
y <- 0
cat("k3( )=",k3( ),"\n",sep="")
# (A)6. (B)9. (C)13. (D)7. (E)8.
A133. [  ]
Q134. # Inner closure
x <- 8
y <- 1
z <- 0
k1 <- function(x) {
  k3 <- function(x) {
    x <<- 4
    y <<- 2
    z <<- 6
    k2 <- function(y) {
      y * 10
    }
    k2(x) + 1
  }
  k3(x) * z
}
cat("k1(0)=",k1(0),"\n",sep=’’)
# (A)250. (B)243. (C)247. (D)246. (E)252.
A134. [  ]
Q135. # Triple nested function/k2
k2 <- function(u) {
  k1 <- function(u) {
    k3 <- function(u) {
      u + 1
    }
    k3(u) + 4
  }
  k1(u) + 4
}
cat("k2(9)=",k2(9),"\n",sep="")
# (A)19. (B)20. (C)15. (D)17. (E)18.
A135. [  ]
Q136. # Double nested function using global u/k1
u <- 5
k1 <- function( ) {
  function( ) u
}
u <- 2
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)5. (B)4. (C)2. (D)1. (E)0.
A136. [  ]
Q137. ## Global x returned
x <- 7
n <- function(x){
  x <<- x + 9
}
cat("n(3)=",n(3),’\n’,sep="")
# (A)12. (B)11. (C)13. (D)10. (E)14.
A137. [  ]
Q138. # Single function, lexical scoping/k3
z <- 5
k3 <- function( ) {
  z + 3
}
z <- 8
cat("k3( )=",k3( ),"\n",sep="")
# (A)8. (B)13. (C)9. (D)11. (E)12.
A138. [  ]
Q139. # Local x returned
x <- 3
n <- function(x){
  x <- x + 8
}
cat("n(4)=",n(4),’\n’,sep="")
# (A)10. (B)12. (C)14. (D)13. (E)9.
A139. [  ]
Q140. # Double nested function using global y/k1
u <- 8
y <- 4
k1 <- function(u) {
  function( ) {
    u + y # which u is used (global u or k1(u))?
  }
}
u <- 0
y <- 2
cat("k1(9)( )=",k1(9)( ),"\n",sep="")
# (A)11. (B)9. (C)14. (D)8. (E)17.
A140. [  ]
Q141. ## Global z returned
z <- 7
n <- function(z){
  z <<- z + 0
```

```
}
cat("n(2)=",n(2),’\n’,sep="")
# (A)1. (B)4. (C)8. (D)2. (E)3.
A141. [  ]
Q142. # Inner closure
u <- 4
x <- 0
y <- 7
k2 <- function(u) {
  k3 <- function(u) {
    u <<- 1
    x <<- 2
    y <<- 6
    k1 <- function(x) {
      x * 10
    }
    k1(u) + 0
  }
  k3(u) * y
}
cat("k2(7)=",k2(7),"\n",sep=’’)
# (A)61. (B)56. (C)59. (D)57. (E)60.
A142. [  ]
Q143. # Double nested identity function/k2
u <- 2
k2 <- function(u) { # u unused
  function(u) u
}
u <- 7
cat("k2(u)(u)=",k2(u)(u),"\n",sep="")
# (A)7. (B)3. (C)13. (D)9. (E)6.
A143. [  ]
Q144. # Global modified, local returned.
u <- 9
z <- 1 # unused
k2 <- function(z){
  u <<- z + 4
  u # local u returned
}
cat("k2(3)=",k2(3),’\n’,sep="")
# (A)4. (B)9. (C)8. (D)7. (E)6.
A144. [  ]
Q145. # Double nested function using global x/k2
x <- 0
k2 <- function( ) {
  function( ) x
}
x <- 5
cat("k2( )( )=",k2( )( ),"\n",sep="")
# (A)12. (B)3. (C)2. (D)7. (E)5.
A145. [  ]
Q146. # Local x returned
x <- 10
n <- function(x){
  x <- x + 2
}
cat("n(7)=",n(7),’\n’,sep="")
# (A)10. (B)7. (C)12. (D)11. (E)9.
A146. [  ]
Q147. # Closure substractor
```

```
k1 <- function(z=9) {
  function(x) {
    x - z
  }
}
cat("k1(1)(9)=",k1(1)(9),"\n")
# (A)2. (B)-5. (C)8. (D)9.
A147. [  ]
Q148. # Global modified, local returned.
y <- 1
x <- 9 # unused
k3 <- function(x){
  y <<- x + 4
  y # local y returned
}
cat("k3(6)=",k3(6),’\n’,sep="")
# (A)8. (B)11. (C)7. (D)10. (E)9.
A148. [  ]
Q149. # Single function, lexical scoping/k1
y <- 9
k1 <- function(y) {
  y + 3
}
y <- 4
cat("k1(y)=",k1(y),"\n",sep="")
# (A)6. (B)5. (C)8. (D)0. (E)7.
A149. [  ]
Q150. # Triple nested global k1.
z <- 6
x <- 3
k1 <- function(z) {
  z <- 2
  k3 <- function(u) {
    u <- 5
    k2 <- function(y) {
      y * 2 + x
    }
    k2(u) + 8 + x
  }
  k3(z) * 9 + x
}
cat("k1(3)=",k1(3),"\n",sep="")
# (A)222. (B)221. (C)223. (D)219. (E)218.
A150. [  ]
Q151. # Triple nested global k2.
z <- 0
u <- 4
k2 <- function(z) {
  z <- 2
  k3 <- function(y) {
    y <- 8
    k1 <- function(x) {
      x * 2 + u
    }
    k1(y) + 3 + u
  }
  k3(z) * 10 + u
}
cat("k2(4)=",k2(4),"\n",sep="")
# (A)274. (B)277. (C)272. (D)276. (E)270.
```

```
A151. [  ]
Q152. # Closure substractor
k2 <- function(u=5) {
  function(x) {
    x - u
  }
}
cat("k2(4)(5)=",k2(4)(5),"\n")
# (A)-3. (B)8. (C)1. (D)-6.
A152. [  ]
Q153. # Double nested function using global u/k3
u <- 7
k3 <- function( ) {
  function( ) u
}
u <- 10
cat("k3( )( )=",k3( )( ),"\n",sep="")
# (A)10. (B)12. (C)6. (D)8. (E)11.
A153. [  ]
Q154. # Local y returned
y <- 2
n <- function(y){
  y <- y + 1
}
cat("n(7)=",n(7),’\n’,sep="")
# (A)14. (B)5. (C)8. (D)11. (E)9.
A154. [  ]
Q155. # Nested functions, lexical scoping
k2 <- function(z) {
  k3 <- function(x) {
    return(z * x)
  }
  return(k3)
}
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)61. (B)62. (C)64. (D)60. (E)63.
A155. [  ]
Q156. # Triple nested function/k1
k1 <- function(y) {
  k3 <- function(y) {
    k2 <- function(y) {
      y + 10
    }
    k2(y) + 8
  }
  k3(y) + 8
}
cat("k1(5)=",k1(5),"\n",sep="")
# (A)32. (B)29. (C)31. (D)26. (E)34.
A156. [  ]
Q157. # Double nested function/n
z <- 0
y <- 4
x <- 8
n <- function(u) {
  y <- u
  function(v) {
    z + y + v # which z, y are used?
  }
}
```

```
z <- 6
y <- 3
x <- 10
cat("n(y)(x)=",n(y)(x),"\n",sep="")
# (A)20. (B)17. (C)16. (D)21. (E)19.
A157. [  ]
Q158. # Two functions, lexical scoping
a <- 1
k2 <- function(z) {
  a * z
}
a <- 10
k3 <- function(x) {
  a <- 100
  k2(x)
}
cat("k3(3)=",k3(3),"\n",sep="")
# (A)35. (B)32. (C)31. (D)29. (E)30.
A158. [  ]
Q159. # Double nested identity function/k1
z <- 8
k1 <- function(z) { # z unused
  function(z) z
}
z <- 3
cat("k1(z)(z)=",k1(z)(z),"\n",sep="")
# (A)2. (B)5. (C)1. (D)3. (E)4.
A159. [  ]
Q160. # Double nested identity function/k2
y <- 1
k2 <- function(y) { # y unused
  function(y) y
}
y <- 6
cat("k2(y)(y)=",k2(y)(y),"\n",sep="")
# (A)7. (B)5. (C)9. (D)6. (E)2.
A160. [  ]
Q161. # Two functions, lexical scoping
a <- 1
k1 <- function(u) {
  a * u
}
a <- 10
k2 <- function(y) {
  a <- 100
  k1(y)
}
cat("k2(3)=",k2(3),"\n",sep="")
# (A)28. (B)34. (C)30. (D)26. (E)33.
A161. [  ]
Q162. # Local x returned
x <- 0
n <- function(x){
  x <- x + 2
}
cat("n(5)=",n(5),’\n’,sep="")
# (A)8. (B)9. (C)6. (D)10. (E)7.
A162. [  ]
Q163. # Double nested function/n
x <- 3
```

```
y <- 1
z <- 2
n <- function(u) {
  y <- u
  function(v) {
    x + y + v # which x, y are used?
  }
}
x <- 9
y <- 6
z <- 10
cat("n(y)(z)=",n(y)(z),"\n",sep="")
# (A)24. (B)28. (C)26. (D)25. (E)21.
A163. [  ]
Q164. ## Global z returned
z <- 3
n <- function(z){
  z <<- z + 10
}
cat("n(5)=",n(5),’\n’,sep="")
# (A)16. (B)11. (C)15. (D)13. (E)22.
A164. [  ]
Q165. # Single function, lexical scoping/k3
z <- 2
k3 <- function(z) {
  z + 1
}
z <- 6
cat("k3(z)=",k3(z),"\n",sep="")
# (A)4. (B)10. (C)6. (D)2. (E)7.
A165. [  ]
Q166. # Global modified, local returned.
u <- 6
y <- 10 # unused
k2 <- function(y){
  u <<- y + 2
  u # local u returned
}
cat("k2(5)=",k2(5),’\n’,sep="")
# (A)7. (B)12. (C)9. (D)4. (E)14.
A166. [  ]
Q167. # Single function, lexical scoping/k3
x <- 7
k3 <- function( ) {
  x + 5
}
x <- 1
cat("k3( )=",k3( ),"\n",sep="")
# (A)7. (B)8. (C)6. (D)3. (E)9.
A167. [  ]
Q168. # Double nested identity function/k2
z <- 5
k2 <- function(z) { # z unused
  function(z) z
}
z <- 0
cat("k2(z)(z)=",k2(z)(z),"\n",sep="")
# (A)0. (B)-7. (C)-1. (D)1. (E)-2.
A168. [  ]
Q169. # Global modified, local returned.
```

```
x <- 2
u <- 4 # unused
k2 <- function(u){
  x <<- u + 7
  x # local x returned
}
cat("k2(1)=",k2(1),’\n’,sep="")
# (A)7. (B)6. (C)8. (D)14. (E)4.
A169. [  ]
Q170. # Closure substractor
k3 <- function(u=5) {
  function(z) {
    z - u
  }
}
cat("k3(8)(5)=",k3(8)(5),"\n")
# (A)-3. (B)-10. (C)-4. (D)-7.
A170. [  ]
Q171. # Triple nested global k1.
y <- 2
u <- 8
k1 <- function(y) {
  y <- 2
  k3 <- function(z) {
    z <- 9
    k2 <- function(x) {
      x * 2 + u
    }
    k2(z) + 5 + u
  }
  k3(y) * 10 + u
}
cat("k1(8)=",k1(8),"\n",sep="")
# (A)399. (B)400. (C)397. (D)398. (E)395.
A171. [  ]
Q172. # Single function, lexical scoping/k3
u <- 10
k3 <- function(u) {
  u + 2
}
u <- 8
cat("k3(u)=",k3(u),"\n",sep="")
# (A)14. (B)8. (C)10. (D)7. (E)17.
A172. [  ]
Q173. # Double nested function using global x/k2
x <- 5
k2 <- function( ) {
  function( ) x
}
x <- 2
cat("k2( )( )=",k2( )( ),"\n",sep="")
# (A)7. (B)1. (C)-3. (D)0. (E)2.
A173. [  ]
Q174. # Nested functions, lexical scoping
k2 <- function(z) {
  k3 <- function(y) {
    return(z * y)
  }
  return(k3)
}
```

```
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)61. (B)59. (C)63. (D)62. (E)65.
A174. [  ]
Q175. # Inner closure
x <- 9
u <- 3
y <- 7
k2 <- function(x) {
  k1 <- function(x) {
    x <<- 2
    u <<- 10
    y <<- 1
    k3 <- function(u) {
      u * 10
    }
    k3(x) + 3
  }
  k1(x) * y
}
cat("k2(7)=",k2(7),"\n",sep=’’)
# (A)22. (B)23. (C)26. (D)20. (E)25.
A175. [  ]
Q176. # Inner closure
y <- 2
z <- 0
u <- 1
k2 <- function(y) {
  k3 <- function(y) {
    y <<- 5
    z <<- 6
    u <<- 10
    k1 <- function(z) {
      z * 10
    }
    k1(y) + 0
  }
  k3(y) * u
}
cat("k2(1)=",k2(1),"\n",sep=’’)
# (A)501. (B)496. (C)500. (D)498. (E)502.
A176. [  ]
Q177. # Nested functions, lexical scoping
k2 <- function(x) {
  k3 <- function(z) {
    return(x * z)
  }
  return(k3)
}
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)61. (B)64. (C)62. (D)63. (E)65.
A177. [  ]
Q178. # Closure substractor
k2 <- function(z=3) {
  function(x) {
    x - z
  }
}
cat("k2(6)(3)=",k2(6)(3),"\n")
# (A)0. (B)-3. (C)-9. (D)2. (E)4.
A178. [  ]
```

```
Q179. # Single function, lexical scoping/k1
u <- 10
k1 <- function( ) {
  u + 7
}
u <- 5
cat("k1( )=",k1( ),"\n",sep="")
# (A)13. (B)10. (C)12. (D)15. (E)14.
A179. [  ]
Q180. # Nested functions, lexical scoping
k2 <- function(u) {
  k1 <- function(z) {
    return(u * z)
  }
  return(k1)
}
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)69. (B)68. (C)64. (D)61. (E)63.
A180. [  ]
Q181. # Global modified, local returned.
x <- 10
y <- 3 # unused
k2 <- function(y){
  x <<- y + 0
  x # local x returned
}
cat("k2(6)=",k2(6),’\n’,sep="")
# (A)6. (B)7. (C)5. (D)4. (E)9.
A181. [  ]
Q182. ## Global z returned
z <- 5
n <- function(z){
  z <<- z + 4
}
cat("n(9)=",n(9),’\n’,sep="")
# (A)14. (B)11. (C)10. (D)13. (E)12.
A182. [  ]
Q183. ## Global y returned
y <- 3
n <- function(y){
  y <<- y + 8
}
cat("n(5)=",n(5),’\n’,sep="")
# (A)13. (B)11. (C)5. (D)12. (E)14.
A183. [  ]
Q184. # Inner closure
z <- 1
y <- 5
x <- 4
k2 <- function(z) {
  k1 <- function(z) {
    z <<- 6
    y <<- 9
    x <<- 0
    k3 <- function(y) {
      y * 10
    }
    k3(z) + 5
  }
  k1(z) * x
```

```
}
cat("k2(4)=",k2(4),"\n",sep=’’)
# (A)0. (B)-1. (C)5. (D)1. (E)-2.
A184. [  ]
Q185. # Two functions, lexical scoping
a <- 1
k2 <- function(u) {
  a * u
}
a <- 10
k3 <- function(x) {
  a <- 100
  k2(x)
}
cat("k3(3)=",k3(3),"\n",sep="")
# (A)30. (B)28. (C)33. (D)29. (E)31.
A185. [  ]
Q186. ## Global u returned
u <- 1
n <- function(u){
  u <<- u + 2
}
cat("n(0)=",n(0),’\n’,sep="")
# (A)3. (B)2. (C)0. (D)1. (E)-1.
A186. [  ]
Q187. # Two functions, lexical scoping
a <- 1
k1 <- function(u) {
  a * u
}
a <- 10
k2 <- function(x) {
  a <- 100
  k1(x)
}
cat("k2(3)=",k2(3),"\n",sep="")
# (A)26. (B)30. (C)27. (D)31. (E)29.
A187. [  ]
Q188. ## Global z returned
z <- 0
n <- function(z){
  z <<- z + 6
}
cat("n(10)=",n(10),’\n’,sep="")
# (A)16. (B)17. (C)22. (D)18. (E)9.
A188. [  ]
Q189. # Global modified, local returned.
y <- 0
z <- 4 # unused
k3 <- function(z){
  y <<- z + 6
  y # local y returned
}
cat("k3(3)=",k3(3),’\n’,sep="")
# (A)10. (B)7. (C)8. (D)11. (E)9.
A189. [  ]
Q190. # Triple nested function/k1
k1 <- function(y) {
  k3 <- function(y) {
    k2 <- function(y) {
      y + 4
    }
    k2(y) + 1
  }
  k3(y) + 1
}
cat("k1(6)=",k1(6),"\n",sep="")
# (A)18. (B)11. (C)17. (D)12. (E)8.
A190. [  ]
Q191. # Nested functions, lexical scoping
k2 <- function(y) {
  k1 <- function(x) {
    return(y * x)
  }
  return(k1)
}
cat("k2(7)(9)=",k2(7)(9),"\n",sep="")
# (A)63. (B)64. (C)62. (D)65. (E)59.
A191. [  ]
Q192. # Double nested function using global u/k2
x <- 1
u <- 5
k2 <- function(x) {
  function( ) {
    x + u # which x is used (global x or k2(x))?
  }
}
x <- 10
u <- 7
cat("k2(2)( )=",k2(2)( ),"\n",sep="")
# (A)9. (B)7. (C)11. (D)14. (E)12.
A192. [  ]
Q193. # Double nested function using global z/k1
z <- 8
k1 <- function( ) {
  function( ) z
}
z <- 7
cat("k1( )( )=",k1( )( ),"\n",sep="")
# (A)7. (B)3. (C)8. (D)6. (E)5.
A193. [  ]
Q194. # Inner closure
u <- 0
y <- 4
x <- 8
k2 <- function(u) {
  k1 <- function(u) {
    u <<- 6
    y <<- 3
    x <<- 10
    k3 <- function(y) {
      y * 10
    }
    k3(u) + 4
  }
  k1(u) * x
}
cat("k2(8)=",k2(8),"\n",sep=’’)
# (A)635. (B)640. (C)644. (D)643. (E)642.
A194. [  ]
```

```
Q195. # Nested functions, lexical scoping
k1 <- function(u) {
  k2 <- function(y) {
    return(u * y)
  }
  return(k2)
}
cat("k1(7)(9)=",k1(7)(9),"\n",sep="")
# (A)64. (B)61. (C)66. (D)68. (E)63.
A195. [  ]
Q196. # Two functions, lexical scoping
a <- 1
k3 <- function(z) {
  a * z
}
a <- 10
k1 <- function(y) {
  a <- 100
  k3(y)
}
cat("k1(3)=",k1(3),"\n",sep="")
# (A)26. (B)35. (C)30. (D)31. (E)29.
A196. [  ]
Q197. # Single function, lexical scoping/k3
y <- 5
k3 <- function( ) {
  y + 7
}
y <- 9
cat("k3( )=",k3( ),"\n",sep="")
# (A)19. (B)14. (C)16. (D)17. (E)15.
A197. [  ]
```

Solutions:
Q1/4, Q2/3, Q3/4, Q4/1, Q5/1, Q6/4, Q7/3, Q8/1, Q9/5, Q10/2,
Q11/4, Q12/2, Q13/1, Q14/1, Q15/3, Q16/4, Q17/1, Q18/4, Q19/3, Q20/3,
Q21/4, Q22/5, Q23/4, Q24/5, Q25/1, Q26/5, Q27/3, Q28/2, Q29/2, Q30/3,
Q31/5, Q32/4, Q33/1, Q34/3, Q35/4, Q36/2, Q37/3, Q38/4, Q39/5, Q40/3,
Q41/2, Q42/1, Q43/4, Q44/1, Q45/3, Q46/2, Q47/1, Q48/3, Q49/1, Q50/5,
Q51/2, Q52/2, Q53/3, Q54/3, Q55/1, Q56/2, Q57/4, Q58/5, Q59/2, Q60/3,
Q61/4, Q62/2, Q63/5, Q64/5, Q65/2, Q66/1, Q67/2, Q68/2, Q69/3, Q70/2,
Q71/4, Q72/4, Q73/2, Q74/5, Q75/1, Q76/2, Q77/3, Q78/2, Q79/5, Q80/2,
Q81/5, Q82/3, Q83/3, Q84/4, Q85/2, Q86/3, Q87/1, Q88/2, Q89/5, Q90/4,
Q91/1, Q92/4, Q93/2, Q94/5, Q95/3, Q96/2, Q97/4, Q98/4, Q99/3, Q100/4,
Q101/2, Q102/2, Q103/2, Q104/2, Q105/4, Q106/4, Q107/3, Q108/3, Q109/5, Q110/4,
Q111/5, Q112/5, Q113/4, Q114/5, Q115/1, Q116/3, Q117/4, Q118/2, Q119/2, Q120/2,
Q121/4, Q122/2, Q123/5, Q124/5, Q125/1, Q126/2, Q127/1, Q128/4, Q129/1, Q130/4,
Q131/5, Q132/2, Q133/5, Q134/4, Q135/5, Q136/3, Q137/1, Q138/4, Q139/2, Q140/1,
Q141/4, Q142/5, Q143/1, Q144/4, Q145/5, Q146/5, Q147/3, Q148/4, Q149/5, Q150/4,
Q151/1, Q152/3, Q153/1, Q154/3, Q155/5, Q156/3, Q157/5, Q158/5, Q159/4, Q160/4,
Q161/3, Q162/5, Q163/4, Q164/3, Q165/5, Q166/1, Q167/3, Q168/1, Q169/3, Q170/1,
Q171/4, Q172/3, Q173/5, Q174/3, Q175/2, Q176/3, Q177/4, Q178/2, Q179/3, Q180/5,
Q181/1, Q182/4, Q183/1, Q184/1, Q185/1, Q186/2, Q187/2, Q188/1, Q189/5, Q190/4,
Q191/1, Q192/1, Q193/1, Q194/2, Q195/5, Q196/3, Q197/3.

Quiz on [RScript] created on 20170115 seed=20170115 from hw6/hw6.qin