

```

1: 2: *****
2: 3: ./exercisel/example_if2.f90
3: 4:
4: 5: PROGRAM example_if2
5: 6: ***
6: 7: *** 2nd piece of code to demonstrate the importance
7: 8: *** of ordering and the allowed overlapping of the test ranges
8: 9: *** in an IF construct.
9: 10: ***
10: 11:
11: 12: IMPLICIT NONE
12: 13:
13: 14: INTEGER :: i
14: 15:
15: 16: PRINT*, "Enter an INTEGER"
16: 17: READ*, i !*** Get the user to input i
17: 18:
18: 19: !*** IF construct with no overlapping of test ranges.
19: 20: !*** Therefore ORDERING is NOT important
20: 21:
21: 22: IF (i .EQ. 2) THEN
22: 23:   PRINT*, "Integer = 2"
23: 24: ELSE IF (i .EQ. 7) THEN
24: 25:   PRINT*, "Integer = 7"
25: 26: ELSE IF (i .GT. 2 .AND. i .LT. 7) THEN
26: 27:   PRINT*, "Integer is greater than 2 an less than 7"
27: 28: ELSE IF (i .GT. 7 .AND. i .LE. 10) THEN
28: 29:   PRINT*, "Integer is greater than 7 and less than 11"
29: 30: ELSE
30: 31:   PRINT*, "No Match"
31: 32: END IF
32: 33:
33: 34:
34: 35: END PROGRAM example_if2
35: 36: *****
36: 37: *****
37: 38:
38: 39:
39: 40: *****
40: 41: ./exercisel/example_if.f90
41: 42:
42: 43: PROGRAM example_if
43: 44: ***
44: 45: *** First piece of code to demonstrate the importance
45: 46: *** of ordering and the allowed overlapping of the test ranges
46: 47: *** in an IF construct.
47: 48:
48: 49: IMPLICIT NONE
49: 50:
50: 51: INTEGER :: i
51: 52:
52: 53: PRINT*, "Enter an INTEGER"
53: 54: READ*, i !*** Get the user to input i
54: 55:
55: 56: !*** IF construct with overlapping of test ranges.
56: 57: !*** Therefore ORDERING is important
57: 58:
58: 59: IF (i .EQ. 2) THEN
59: 60:   PRINT*, "Integer = 2"
60: 61: ELSEIF (i .EQ. 7) THEN
61: 62:   PRINT*, "Integer = 7"
62: 63: ELSE IF (i .GE. 2 .AND. i .LE. 8) THEN
63: 64:   PRINT*, "Integer is greater than one and less than nine"
64: 65: ELSE IF (i .GE. 6 .AND. i .LE. 10) THEN
65: 66:   PRINT*, "Integer is larger than five and less than eleven"
66: 67: ELSE
67:

```

```

68: PRINT*, "No Match"
69: END IF
70:
71: END PROGRAM example_if
72: *****
73: *****
74:
75:
76: *****
77: ./exercisel/example_case.f90
78:
79: PROGRAM example_case
80:
81: IMPLICIT NONE
82:
83: INTEGER :: i
84:
85: PRINT*, "Enter an INTEGER"
86: READ*, i !*** Get the user to input i
87:
88: !*** CASE construct NOTE overlapping of ranges in case-selector is
89: !*** NOT permitted. Therefore ORDERING is NOT important
90:
91: SELECT CASE (i)
92:   CASE (2)
93:     PRINT*, "Integer = 2"
94:   CASE (7)
95:     PRINT*, "Integer = 7"
96:   CASE (3:6)
97:     PRINT*, "Integer is greater than 2 an less than 7"
98:   CASE (8:10)
99:     PRINT*, "Integer is greater than 7 and less than 11"
100:   CASE DEFAULT
101:     PRINT*, "No Match"
102: END SELECT
103:
104: END PROGRAM example_case
105:
106: *****
107:

```