# Review: Relational & Logical expressions

1. The following truth table shows various combinations of the values true and false connected by a logical operator. Complete the table by indicating if the result of such a combination is true or false.

| Logical Expression | Result (true or false) |
|---|---|
| true && false | false |
| true && true | true |
| false && false | false |
| true \|\| false | true |
| true \|\| true | true |
| false \|\| false | false |
| !true | false |
| !false | true |

2. If a = 2, b = 4, and c = 6, indicate whether each of the following conditions is true or false:

| | |
|---|---|
| A) (a == 4) \|\| (b >2) | T |
| B) (6 <= c) && (a > 3) | F |
| C) (1 != b) && (c != 3) | T |
| D) (a>=-l) \|\| (a<=b) | T |
| E) !(a > 2) | T |

3. If a = 2, b = 4, and c = 6, is the following expression true or false?
   (b > a) || (b > c) && (c == 5)    True (&& is done before ||)

4. Rewrite the following using the ! operator so that the logic remains the same.
   if (activeEmployee == false)    if (!activeEmployee )

5. Assume the variables x = 5, y = 6, and z = 8. Indicate if each of the following conditions is true or false:

| | |
|---|---|
| A) (x==5) \|\| (y>3) | T |
| B) (7 <= x) && (z > 4) | F |
| C) (2 !=y) && (z !=4) | T |

6. Assume the variables x = 5, y = 6, and z = 8. Indicate if each of the following conditions is true or false:

| | |
|---|---|
| A) (x>=0) \|\| (x<=y) | T |
| B) (z - y) > y | F |
| C) !((z — y) > x) | T |

7. Write an if statement that sets the variable hours to 10 when the flag variable minimum is set.
```
if(minimum)
  hours=10;
```
8. Convert the following conditional expression into an if/else statement.
   q = (x < y)  ?  (a + b) : ( x* 2); //notice the use of the conditional operator
```
if(x<y)
    q=a+b;
else
   q=x*2
```
9. Write a C++ statement that prints the message "The number is valid." if the variable grade is within the range 0 through 100.
```
if (grade >=0 && grade <=100)
   cout <<"The number is valid.";
```
10. Write a C++ statement that prints the message "The number is valid." if the variable temperature is within the range —50 through 150.
```
 if (temperature >=-50 && temperature <=150)
   cout <<"The number is not valid.";
```
11. Write a C++ statement that prints the message "The number is not valid." if the variable hours is outside the range 0 through 80.
```
  if (hours<0 || hours >80)
   cout <<"The number is not valid.";
```
12. Write a C++ statement that displays the strings titlel and title2 in alphabetical order.
```
if(title1<title2)
  cout<<title1 <<" "<<title2<<endl;
else
    cout<<title2 <<" "<<title1<<endl;
```

** With using C-strings, you must replace the above if statement with:

```
if (strcmp(title1, title2) <= 0)
```

**Soft Skills – Discussion item**
Programmers need to be able to look at alternative approaches to solving a problem and at different ways of implementing a solution, weighing the pros and cons of each. Further, they need to be able to clearly articulate to others why they recommend, or have chosen, a particular solution. Discuss the following:

Sometimes either a switch statement or an if/else if statement can be used to implement logic that requires branching to different blocks of program code. But the two are not interchangeable.

A) Under what circumstances would an if / else if statement be a more appropriate choice than a switch statement?

An if/else if is more appropriate than a switch statement when all test expressions do not involve the same variable or when test expressions need to test more than one condition, work with non-integer values, or use relational operators that test for something other than equality.

B) Under what circumstances would a switch statement be a more appropriate choice than an if/else if statement?

A switch statement is more appropriate than an if/else if statement when all tests are comparing a variable for equality with just 1 or a small set of integer values. It is a particularly useful construct to use when you want to utilize the "fall through" feature to carry out more than 1 set of actions when a particular condition is true.

C) Under what circumstances would a set of nested if/else statements be more appropriate than either of the other two structures?

A set of nested if/else statements is more appropriate than either of the other two constructs when the test conditions that determine the actions to be carried out do not fall into a neat set of mutually exclusive cases. For example, if one condition is true, then which set of actions you wish to take may depend on the outcome of a second test.

Try to come up with at least one example case for each of the three, where it is the best way to implement the desired branching logic.