The AND special form
The AND special form is one of the logical operators. The arguments to this form are booleans. The AND inputs are evaluated only until one evaluates to false and then the return is false.
Another way to say this is that an AND evaluates to true iff all its inputs are true. No more evaluation occurs after a false expression is found (ie, short-circuit evaluation).

For the following uses of and, evaluate the result and underline the parts of the expression that are not evaluated:
(and (< 5 6) (> 9 8)) \rightarrow (and #t #t) \rightarrow #t
(and (> x 1) (zero? (remainder x 2)))
  if x = 2, evaluates to (and #t #t) \rightarrow #t
  if x = 1, evaluates to (and #f) \rightarrow #f and the expression with zero? is not evaluated

The OR special form
The OR special form is one of the logical operators. The arguments to this form are booleans. The OR inputs are evaluated only until one evaluates to true and then the return is true.
Another way to say this is that an OR evaluates to false iff all its inputs are false. No more evaluation occurs after a true expression is found (ie, short-circuit evaluation).

For the following uses of or, evaluate the result and underline the parts of the expression that are not evaluated:
(or (< 5 6) (> 9 8)) \rightarrow (or #t #t) \rightarrow #t
(or (> x 1) (zero? (remainder x 2)))
  if x \geq 2, evaluates to (or #t #t) \rightarrow #t and the zero? expression is not evaluated
  if x = 1, evaluates to (or #f #f) \rightarrow #f

The NOT function
The not function takes 1 boolean input and returns the negation of its input, for example:

(not (\geq 8 7 6 5 3)) \rightarrow (not #t) \rightarrow #f
(not (and #t #f)) \rightarrow (not #f) \rightarrow #t
The IF special form

The IF special form is the basic 2-way decision statement. It is written like this:

```
(if (> x p) (+ p 1) (- x 2))
```

There are usually three inputs to the if. The first is a **predicate expression**. The second is the **result returned if the predicate evaluates to #t** and the third is the **result returned if the predicate evaluates to #f**. What would the if above return if 
\[ x = 8 \] and \[ p = 4 \]?

When explaining the evaluation of an if statement, I often refer to the "true part" (the second input) and the "false part" (the third input). While you can do multi-way decisions using a series of IF statements, the meaning of the statement may become obscured. There is a better way to do multi-way decisions, and that is the COND special form.

The COND special form

The COND special form is the multi-way decision statement. The general form is:

```
(cond
  [[predicate) return]  [[predicate) return]
  ...  ...  
  [[predicate) return])  [else return])
```

The dots indicate that the cond may have an arbitrary number of clauses (including only 2).

Example COND expressions:

```
(cond
  [(>= n 90) 'A]
  [(>= n 80) 'B]
  [(>= n 70) 'C])
```

Each part of the cond inside [']s is called a clause. Each clause can have only 2 parts, a predicate and a return value. Only 1 cond clause return value is ever returned.

For the first condition that evaluates to true, Scheme evaluates the corresponding answer, and the value of the answer is the value returned by the cond-expression. If the last condition is else and all other conditions fail, the answer for the cond is the value returned in the else clause.

Eg, Write a function that takes one argument, an integer representing a student’s score on an exam. For a score >= 90, the return is 'A, for a score >= 80, the return is 'B, for a score >=70, the return is 'C, for a score >= 60 the return is 'D, else 'F
The order of questions is important. For example, what would be returned if we used the cond on the left for input \(n=0\), \(n=90\), or \(n=25\)?

What would be returned in the cond on the right for an input of \(n=90\), \(n=80\), or \(n=40\)?

\[
\begin{align*}
\text{(cond} & \quad \text{(cond)} \\
[\langle n \ 30 \rangle \ 5] & \quad [\langle n \ 70 \rangle \ 'A'] \\
[\langle n \ 20 \rangle \ 50] & \quad [\langle n \ 80 \rangle \ 'B'] \\
[\langle n \ 10 \rangle \ 1)] & \quad \text{else} \quad 'C')
\end{align*}
\]