Logical Operators

As we saw in the truth table, a Boolean expression that uses logical operators can be as simple as evaluating two boolean values:

```
bool answer = true && false; // evaluates to False
```

In this case, we're saying that `answer` is equal to the evaluation of `true` AND `false`. According to the truth table, `answer` will return False.

But more often, Boolean expressions are extremely complex. Rather than determining if one relationship is true or false, we can evaluate several expressions by connecting them with logical operators and then determining the validity of the overall expression.

```
bool answer = (9 < 3) || (100 < 45); // evaluates to False
bool another = ((3439 > 40) && (1 < 3)) || answer; // evaluates to True
```

We can use logical values to start chaining logical statements. Let's start by finding the value of `answer`.

First, the computer will evaluate each comparison statement:

```
bool answer = (9 < 3) || (100 < 45)
```

Both of these statements evaluate to false:

```
bool answer = false || false
```

Since both statements evaluate to false and we're using an OR operator, the overall expression will return false:

```
bool answer = false
```

Now we can start evaluating the value of `another`. Again, we'll start by evaluating the comparison statements:

```
bool another = ((3439 > 40) && (1 < 3)) || answer
```

Both statements evaluate to true:

`bool another = (true && true) || answer`

Since both statements evaluate to true and we're using an AND operator, the overall expression returns true:

`bool another = true || answer`

Since we already know that `answer` is false and we're evaluating it with a true value using an OR operator `another`, it will return true:

`bool another = true`.

☑Instructions

1.

You and your friend are planning a trip together and are trying to decide where to go. You each have specific criteria that it needs to fulfill. You want it to have a beach and city, whereas your friend wants beach or hiking. Your current pick is Barcelona, which is a city that has a beach. Will both you and your friend be happy?

Create a `bool` variable named `yourNeeds`. Write a logical comparison that captures your criteria.

Hint

Since you need both criteria to be fulfilled, use the `&&` operator:

```
bool peanutButter = true;
bool jelly = true;

bool sandwich  = (peanutButter && jelly); // evaluates to true
```

2.
Create a `bool` variable named `friendNeeds`. Write a logical comparison that captures their criteria.

Hint

Since only one criteria needs to be fulfilled, use the `||` operator:

```
bool peanutButter = true;
bool marshmallow = true;

bool sandwich  = (peanutButter || marshmallow); // evaluates to true
```

3.
Write a logical comparison that compares `yourNeeds` and `friendNeeds` and save it to `tripDecision`.

Print `tripDecision` to the console. Will you agree on Barcelona? You should only go if it satisfies both your needs and your friend's needs.

What about Portbou, a smaller town (not a city) with a beach but no hiking?

Hint

Since you need both criteria to be fulfilled, use the `&&` operator:

```
bool peanutButter = true;
bool jelly = true;

bool sandwich  = (peanutButter && jelly); // evaluates to true
```