

## TDA/DIT 594 - Category Partition Method Activity

You have created a utility intended to find all instances of a **pattern** in a **file**.

### `find(pattern, file)`

This pattern can contain spaces and quotes. For example:

```
find("john",myFile)
```

Finds all instances of john in the file

```
find("john smith",myFile)
```

Finds all instances of john smith in the file

```
find("“john” smith",myFile)
```

Finds all instances of "john" smith in the file

Use the category-partition method to identify a pool of valid test specifications.

1. Identify the **choices** that you control when testing.
2. For each choice, identify a set of **representative values** that could lead to different outcomes of the function.
3. Impose constraints on the choices to reduce the pool of test cases.
  - a. **error** constraints identify values that should result in an error no matter what other values they are paired with.
  - b. **single** constraints identify values that should result in normal execution, but should be tried once because they have the potential for error or strange behavior.
  - c. **if-constraints** identify values that should only be used if other choices are set to particular values ("if choice X = THIS, then choice Y = THAT)