Note: The below regular expressions should utilize metacharacters, character classes, grouping, alternation, and quantifiers where applicable. Also, when creating your regular expressions, please use the notation from the lectures: */regular expression/*

1. Write a regular expression that matches your name

```
/[jJ]arred/
```

2. Write a regular expression that matches the words **Then** and **Than** (use a character class)

```
/Th[ea]n/
```

3. Write a regular expression that matches the sequence **yes** or the sequence **no**

```
/(yes)|(no)/
```

4. Write a regular expression that matches at least one d followed by any number of f's.

```
/d+f*/
```

5. Write a regular expression that matches the structure of a social security number

```
/[0-9]{3}-[0-9]{2}-[0-9]{4}/
```

6. Write a regular expression that matches all names that begin with a **Mc** (case sensitive)

```
/^[Mc].*/
```

7. Write a regular expression that will match all C++ files (files/names that end in a **.c** or **.cc**., don't worry about cpp)

```
/.*\.(c+)/
```

8. Write a regular expression that matches words that contain a sequence of vowels (**aeiou)** (e.g. jjjakkkollllu, akkkku, jallekkillllu I will be lenient on this one, a bit tricky))

```
/.*[aeiou]+.*/
```

9. Write a regular expression that matches words that begin and end with the letter **b** (e.g bob, bulb, etc)

```
/^b.*b$/
```

10. For the below regular expression, give two examples that match the expression and two examples that do not match. **Your examples that match and do not match the expression must include the characters x and y. The entire character set is x and y.**

```
/x+y+/
```

```
matches: xy
matches: xxyy
doesn't match: yx
doesn't match: yyxx
```

11. For the below regular expression, give two examples that match the expression and two examples that do not match.

```
/[a-f]z?/
```

```
matches: az
matches: fz
doesn't match: afzz
doesn't match: bbz
```