report.md

If you were permitted to use Yacc's built-in operator precedence support for binary operations, what parts of your specification would change and why? A short answer of a few sentences and a few small code snippets is all I'm looking for. Please do not write an entire separate specification. Information you need to answer this question is easily found in Yacc manual

1. As found from Bison documentation, we know that operator precendence can be specified with the declarations %left, %right, and %noassoc.

This means that for any given set of operations, such as

```
op1 x op2 y op3
```

We can define the precendence order of x and y using the above precedence declarations. Using %noassoc will mean that this input is invalid, there is no precedence between the two operators. Using %left will group the first two operations together and execute them first. Using %right will group the second two operations together and execute those instead.

This allows us to very quickly and easily define the precedence between the groups of operators.

An example of using this style of precedence is by doing something such as

```
%left '+' '-'
```

This tells Yacc that addition needs to be executed before subtraction.