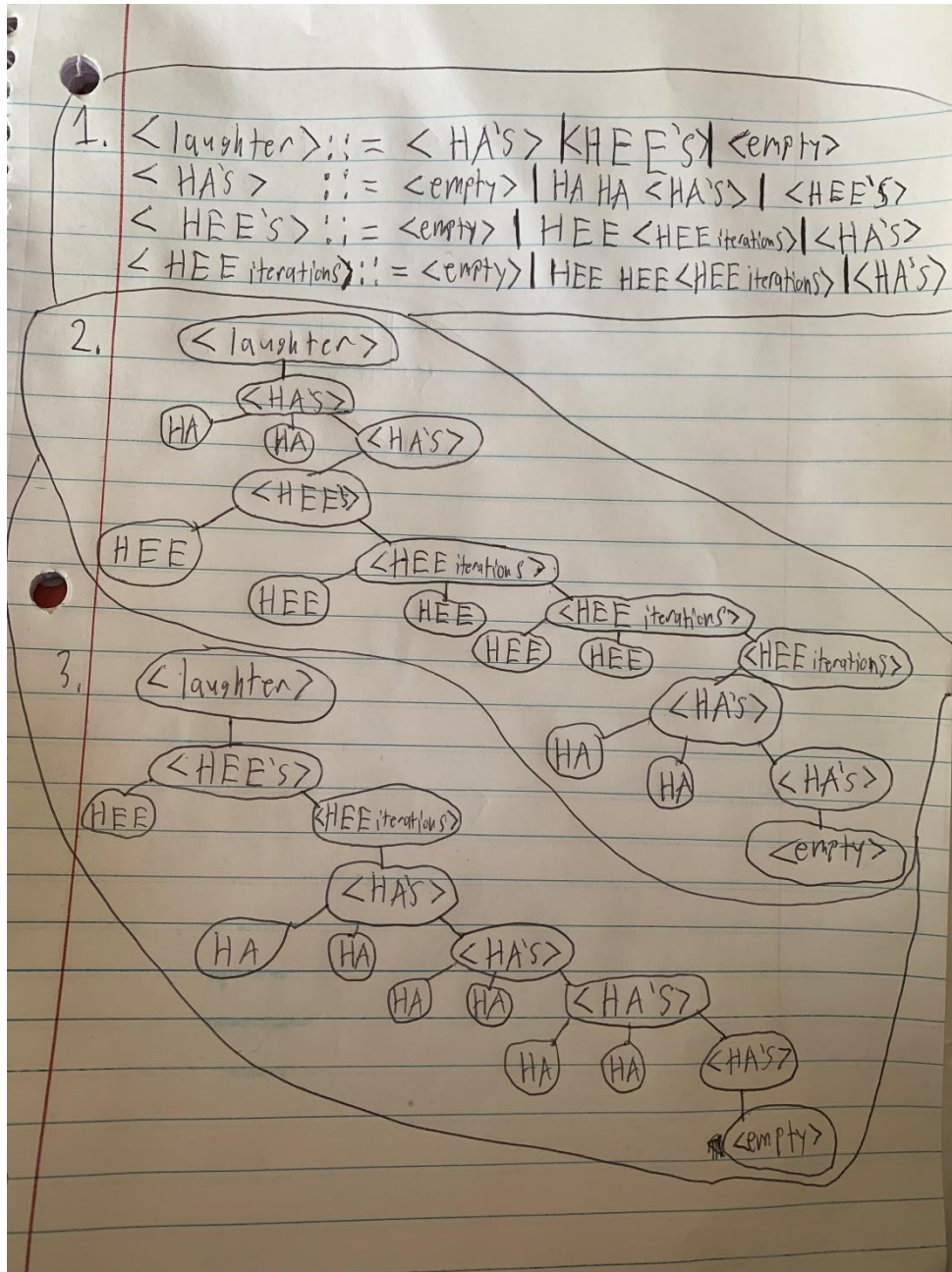# BNF Assignment (Problem Set #1):
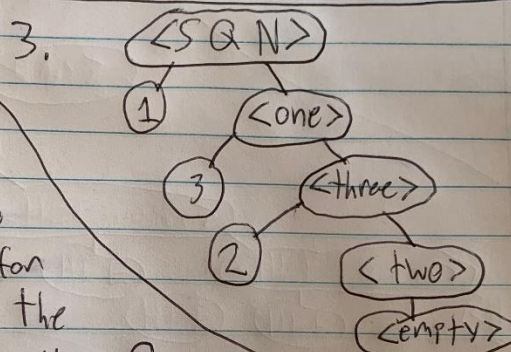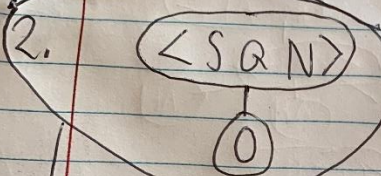
Abstract: In this assignment we were asked to create BNF grammars based on the given language, draw BNF trees and define/explain BNF to a freshman computer science major.

Problem 1 – Laughter

Problem 2 – SQN

1. <SQN> ::= 0 | 1<one> | 2<two> | 3<three>
   <one> ::= <empty> | 2<two> | 3<three> | 0<zero>
   <two> ::= <empty> | 1<one> | 3<three> | 0<zero>
   <three> ::= <empty> | 1<one> | 2<two> | 0<zero>
   <zero> ::= <empty> | 1<one> | 2<two> | 3<three>

2. <SQN>
      |
      0

3. <SQN>
   1   <one>
   3   <three>
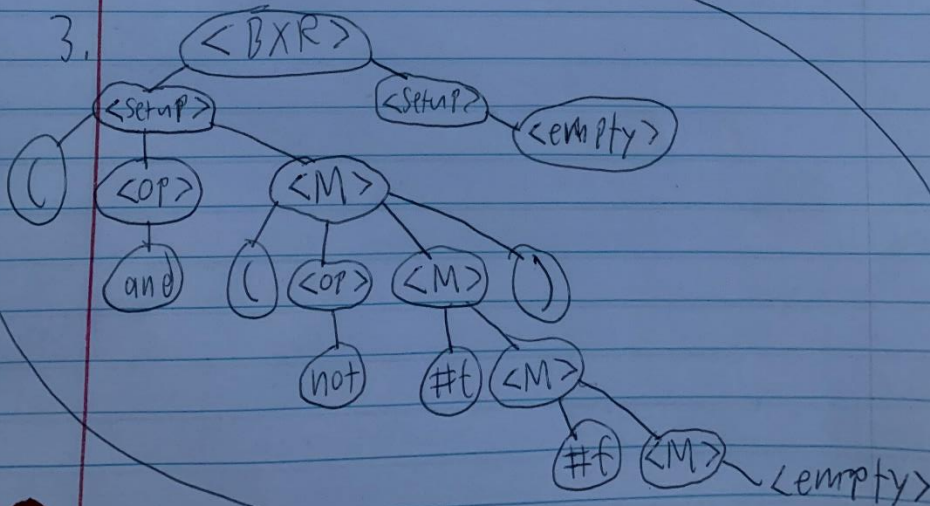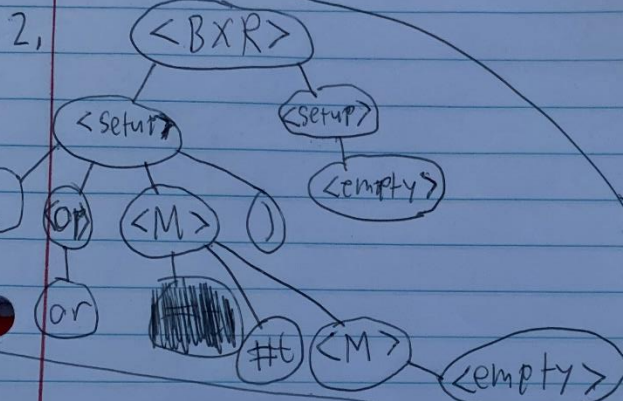   2   <two>
              <empty>

4. When trying to create a parse tree for the number 1223 the problem is the repeating 2. When getting to the first 2, it is physically impossible in this language to produce another 2 as an output. In this language you can only produce a 1, 3, 0 or <empty> after producing a 2.
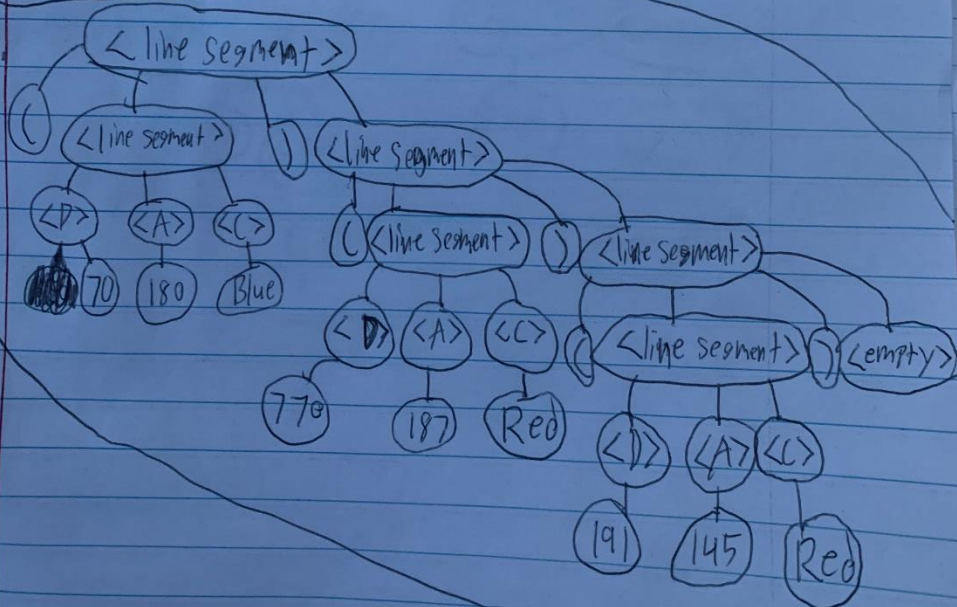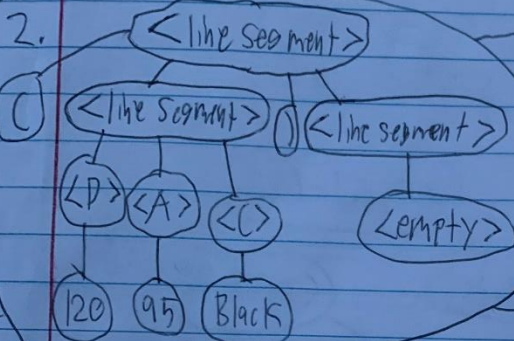
Problem 3 – BXR

1.
```
<BXR> ::= #t | #f | <setup><setup>
<op> ::= and | or | not
<setup> ::= <setup><setup> | (<op><M>) | <empty>
<M> ::= (<op><M>) | #t<M> | #f<M> | <empty>
```


2.

3.

Problem 4 – LSS

1. `<line segment> ::= <empty> | (<line segment> <line segment>) | <D><A><C>`
   `<C> ::= Red | Blue | Black`

2.
```
              <line segment>
        ( <line segment> ) <line segment>
        <D> <A> <C>        <empty>
        120  95  Black
```

3.
```
              <line segment>
        ( <line segment> ) ( <line segment> )
        <D>  <A> <C>        ( <line segment> ) <line segment>
        70   180 Blue       <D> <A> <C>        ( <line segment> ) <empty>
                            770  187  Red       <D> <A> <C>
                                                 91   145  Red
```

Problem 5 – M-Lines

1 <M-Lines> ::= <empty> | < M-Lines> <M-Lines> |
  <M-Lines> ::= | <event> <M-Lines> | LP <M-Line> RP |
  <M-Lines> ::= | RP <M-Lines> LP | X2 < M-Lines> S2
  <M-Lines> ::= | S2 <M-Lines> X2
  <event> ::= play | Rest

2

Continued…

Problem 6 – What is BNF?

B.N.F. stands for Backus Naur Form and is used in order to express the syntax of a programming language. Learning Backus Naur Form can offer deep insight for a computer science by laying out the tools and rules that you must adhere to in order for progress and solidify understanding between the computer scientist and the tool they are using. BNF makes use of tokens, nonterminal, productions and of course the start symbol.