(Please ensure this uses only tokens detected in your P1, no exceptions. Note that some legal tokens will pass the scanner but will always lead to an error here as they are not part of the BNF, that is I decided not to use them)

<program> -> <vars> **main** <block>

<block> -> **{** <vars> <stats> **}**

<vars> -> empty | **declare Identifier := whole ;** <vars>

<expr> -> <N> **-** <expr> | <N>

//<N> -> <N> / <A> | <N> + <A> | <A>

<N> -> <A> <X>

<X> -> / <A><X> | +<A> <X> | empty

<A> -> <M> * <A> | <M>

<M> -> **%** <M> | <R>

<R> -> **(** <expr> **)** | **Identifier** | **Integer**

<stats> -> <stat> <mStat>

<mStat> -> <stat> <mStat> | empty

<stat> -> <in> **;** | <out> **;** | <block> | <if> **;** | <loop1> **;** | <loop2> ; | <assign> ; | <goto> ; | <label> ;

<in> -> **listen Identifier**

<out> -> **yell** <expr>

<if> -> **if [** <expr> <RO> <expr> **] then** <stat>

<loop1> -> **repeat [** <expr> <RO> <expr> **]** <stat>

<loop2> -> **repeat** <stat> **until [** <expr> <RO> <expr> **]**

<assign> -> **assign Identifier =** <expr>

<RO> -> **>=** | **<=** | **==** | **...** (three tokens here) | **!=**

<label> -> **label Identifier**

<goto> -> **portal Identifier**