PROJECT TASK SHEET:

The *Intermediate Code*, which your compiler can already generate (as per previous specification sheet) *is already very similar to the form which we want to finally achieve* in order to make everything executable.

- **Your final output file will contain executable code** in the ancient **BASIC** Syntax, which you can see at **>>>** https://en.wikipedia.org/wiki/BASIC

To **transform** the Abstract Intermediate Code into executable BASIC, only the following few steps of **post-processing** are needed:

- Consecutive **Line-Numbers need to be generated for each line** of the un-numbered Intermediate Program. (By the way, it is "traditional" BASIC "style" to generate the Line-Numbers in steps of 10, 20, 30, etc..., although normal numbering 1,2,3, etc..., is also possible). *Thereafter:*
- Wherever **REM Lx** is a Label-Line in the *un*-numbered Intermediate Code, *and* wherever **GOTO Lx** is a jump command to that label **Lx** in the *un-_numbered Intermediate Code, _and* wherever **LineNumber REM Lx** is the same Label-Line in the **numbered** Code, the command **GOTO Lx** from Intermediate Code *gets replaced by* **GOTO LineNumber** in the executable BASIC Code.
- In a *similar manner*, **THEN Lx** in the un-numbered Intermediate Code *gets replaced by* **THEN LineNumber** in the executable BASIC Code.

The following **Example** provides some **illustration**:

Un-numbered Intermediate Code	*Line-numbered executable BASIC Code*
some_command	130 *some_command*
REM L0	**140 REM L0**
IF V30=V31 **THEN L1**	150 IF V30=V31 **THEN 180**
some_other_command	160 *some_other_command*
GOTO L0	170 GOTO **140**
REM L1	**180 REM L1**
yet_another_command	190 *yet_another_command*

On the **Internet** you can find many **Online-BASIC-Emulators** which enable you to really **RUN a BASIC Program** which your successful Semester-Project-Compiler has automatically generated from a correct **SPL** Input Program : **The Feeling of Success is GREAT :)**

- And now : **HAPPY CODING :)**