**GROUP\_1\_README**

**PHASE 1**

**How to execute:**

* Put input.asm ,binary\_form.txt , type.txt, registers.txt and phase1.cpp together in same file.
* Compile in terminal using command “g++ phase1.cpp”.
* Execute using “.\a.out” in Ubuntu and “a” in windows { before running “phase2.cpp” }.
* “ machine\_code.mc ” created as output.

This phase consist of taking input instructions and data from file and change to machine code in file for phase-2:

**Work Split:**

1. Rahul Dhruv (2017CSB1100):

* Input-Output file handling.
* Branch and .data part as well as .text part.
* Changing instruction and data to machine code.
* Reading all type of instructions type and define actions corresponding to it.
* Few errors are supported for error handling.
* Correct instruction input and differentiation of similar instructions with different execution like “lw”.

2. Shrikant Saxena (2017CSB1111):

* Preparing of type and initial binary format file of instructions.
* Preparing of register input file
* Testing and Debugging
* GUI work in progress
* Integrating raw code in Qt based application
* Support for Syntax Analysis (in progress)

**TERMS ( i.e. limitations):**

* “. word” and “.byte” takes integer as input only not hexagonal.
* keep a space or a line after file end command “fall\_through” of input.asm
* don’t put comma ( , ) between registers input ,only space is enough.
* write .text and .data (to enhance readability).
* x0 can be overwritten , take care of that in the assembly code . eg. don’t use jal xo, label will yield a wrong result.
* Input string in .data should not contain “ ” ,instead directly write string. I.e. abcd (not “abcd”).
* Comments not supported yet.

:---------: