Practical 04

Assembly Language

THIS IS A PROCTORED PRACTICAL

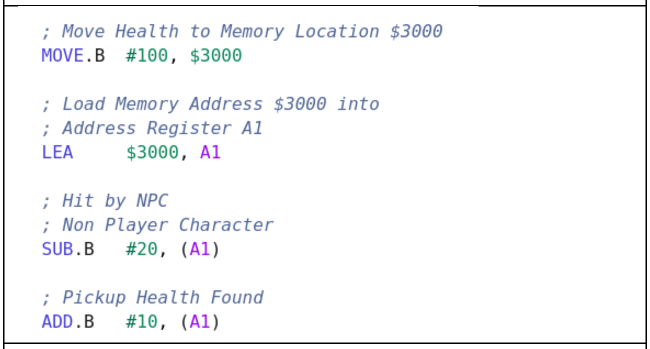
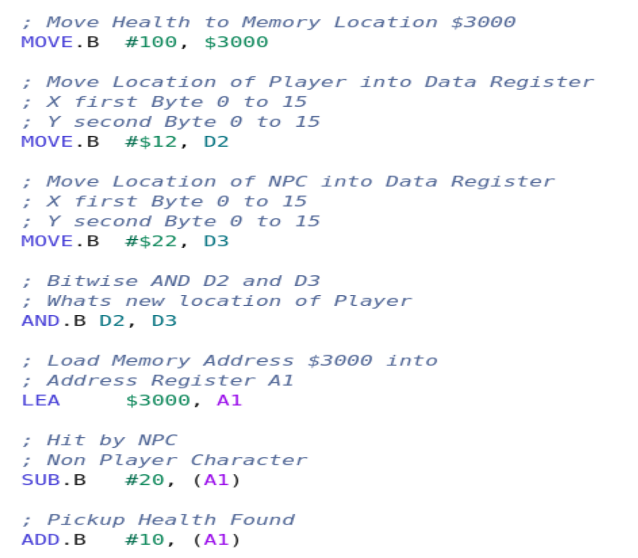
YOU MUST SHARE YOUR SCREEN SO YOUR PARTICIPATION IN THIS PRACTICAL CAN FULLY INVIGILATED

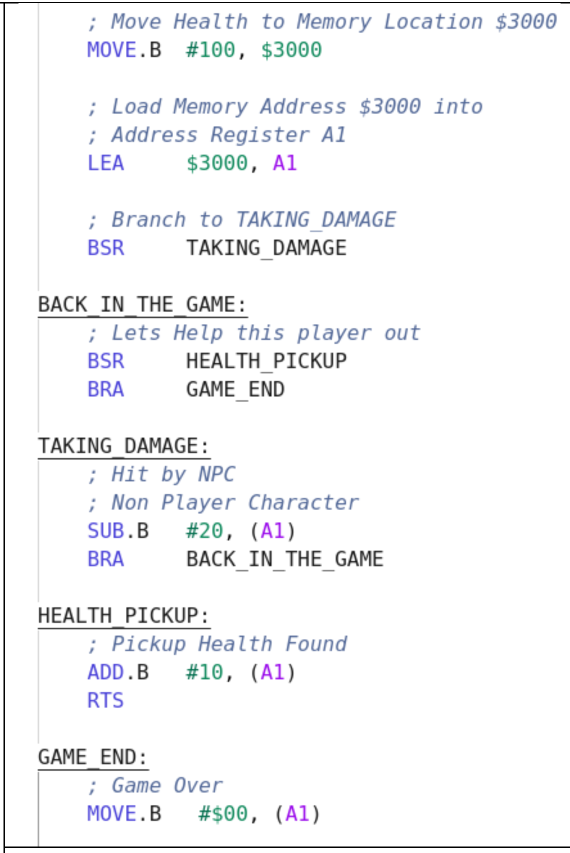
1. Create a Github repository "Assembly_and_C"
2. Create a sub directory PRACTICAL_##
3. Add Github link to CA Spreadsheet
   e.g https://STUDENTID.github.com/Assembly_and_c/PRACTICAL_##
4. Invite Lab Supervisors including **MuddyGames** as a collaborators
5. Go to designated group to complete practical
6. Upload completed Practical files to Github repository

NOTE: Use of EASy68K editor and emulator allowed, use of internet allowed, use of slide deck(s) allowed. Installer located here http://www.easy68k.com/

Create a unique file ***e.g. practical_##_part#.X68*** for each practical section below.

Objective Understand and utilise Arithmetic, Logic Operations and BSR and BRA Branching**:**

1	Create a new 68K project and name the file ***practical_04_part1.X68*** Edit compile and execute the code across and observe while debugging and contents of memory. Examine and note contents of address registers and memory. Review questions, what do the address register mean and what is stored in memory and why?	 ; Move Health to Memory Location $3000 MOVE.B #100, $3000 ; Load Memory Address $3000 into ; Address Register A1 LEA $3000, A1 ; Hit by NPC ; Non Player Character SUB.B #20, (A1) ; Pickup Health Found ADD.B #10, (A1)
2	Create a new 68K project and name the file ***practical_04_part2.X68*** Edit compile and execute the code across and observe while debugging and contents of memory. Examine and note contents of address registers and memory. Write examples for OR, NOT and EOR (Exclusive OR) Review questions, what do the logic instructions mean, what Addressing Modes are used?	 ; Move Health to Memory Location $3000 MOVE.B #100, $3000 ; Move Location of Player into Data Register ; X first Byte 0 to 15 ; Y second Byte 0 to 15 MOVE.B #$12, D2 ; Move Location of NPC into Data Register ; X first Byte 0 to 15 ; Y second Byte 0 to 15 MOVE.B #$22, D3 ; Bitwise AND D2 and D3 ; Whats new location of Player AND.B D2, D3 ; Load Memory Address $3000 into ; Address Register A1 LEA $3000, A1 ; Hit by NPC ; Non Player Character SUB.B #20, (A1) ; Pickup Health Found ADD.B #10, (A1)
3	Create a new 68K project and name the	

	file ***practical_04_part3.X68***
	
	Edit compile and execute the code across and observe while debugging and contents of memory.
	
	Examine and note contents of address registers and memory.
	
	Review questions, what are the Branch Instructions useful for BRA and BSR, what Addressing Modes are used?
		
		`    ; Move Health to Memory Location $3000`
		`    MOVE.B  #100, $3000`
		
		`    ; Load Memory Address $3000 into`
		`    ; Address Register A1`
		`    LEA     $3000, A1`
		
		`    ; Branch to TAKING_DAMAGE`
		`    BSR     TAKING_DAMAGE`
		
		`BACK_IN_THE_GAME:`
		`    ; Lets Help this player out`
		`    BSR     HEALTH_PICKUP`
		`    BRA     GAME_END`
		
		`TAKING_DAMAGE:`
		`    ; Hit by NPC`
		`    ; Non Player Character`
		`    SUB.B   #20, (A1)`
		`    BRA     BACK_IN_THE_GAME`
		
		`HEALTH_PICKUP:`
		`    ; Pickup Health Found`
		`    ADD.B   #10, (A1)`
		`    RTS`
		
		`GAME_END:`
		`    ; Game Over`
		`    MOVE.B   #$00, (A1)`

4	Create a new 68K project and designate the file as ***practical_04_part4.X68***. Perform tasks such as editing, compiling, and executing the code according to the **Specification**. During debugging, closely monitor the contents of the memory. Inspect the stored values in memory and adjust values and their locations within the memory. Review questions what Addressing Modes are used?	**4 Specification**: Complete a simple game that uses • Data Registers • Address Registers • Arithmetic Operations (ADD and SUB) • Logical Operation (AND, OR and EOR) • Branch Instructions (BRA and BSR) Declaring initial data such as Player Health at 100% or $64 (Hex equivalent) Use your own examples, such as typical game data; • Player Points • Player Health • Player X and Y Position • Boss Health • Boss X and Y Position Modify the Data Values during programme operation
5	Complete Practical Quiz which will be provided by Lab Supervisor	

Practical 04

Assembly Language

Demonstrate completed assembly files at the end of the LAB and ensure it has been checked

Student Name		**Student Number**	
Date		**Checked**	