TEAM NUMBER ..... **TEAM GRADE .....**
NAME ....
....

Marks split between team members: □ evenly (each member gets the team grade)
□ unevenly (each member's grade is moderated by peer review)

FEEDBACK FOR
AC21009.3 MANCHESTER BABY SIMULATOR + ASSEMBLER

This is intended as a rough guide only. Note that not all aspects carry equal value.

Your simulator program: **…… out of 50**

Written in:

□ C++ □ C □ Other …….

The code which you submitted: □ ran OK / robust ☺ □ ran with difficulty □ would not run ☹

Code quality:

□ Appears robust overall ☺ □ Some bugs noted □ Very buggy / not working ☹

Program structure	□ very good	□ good	□ poor	/3
Comments within code	□ good comments	□ some comments	□ few/no comments	/2

Functionality:

□ Less than minimal □ Minimal □ Partial □ Complete □ Complete + extensions

Features:

Machine code program is:	□ hard coded	□ from file	□ handles multiple programs		/4
Display uses:	□ text	□ text-based graphics	□ colour		/3
Display content shows:	□ memory	□ CI	□ PI	□ accumulator □ STOP	/5
Decode commentary shows:	□ mnemonics	□ binary	□ operand	□ none ☹	/3
Display is printed:	□ at start	□ at end	□ after each cycle (best)		/3
Fetch-execute cycle demonstration:	□ good	□ OK	□ poor		/4
Fetch-execute cycle runs:	□ in one go	□ step-by-step ☺			/1
Decodes instructions correctly:	□ all	□ most	□ some	□ none ☹	/3
Executes instructions correctly:	□ all	□ most	□ some	□ none ☹	/3
Runs machine code example OK:	□ yes	□ no	□ partly ……		/2

Extensions:

Extended memory:	□ no	□ register width	□ # of addresses	/3
Extended instruction set:	□ no	□ yes ……..….		/2
Extended addressing mode(s):	□ no	□ yes ……..….		/2
Extra machine code program(s):	□ no	□ yes ……..….		/3
High-resolution graphics display:	□ no	□ yes		/2
□ Other useful features noted:				/2

TEAM NUMBER .....

Your assembler program: **…… out of 30**

□ code integrated with simulator

Written in:

□ C++ □ C □ Other …….

The code which you submitted: □ ran OK / robust ☺ □ ran with difficulty □ would not run ☹

Code quality:

□ Appears robust overall ☺ □ Some bugs noted □ Very buggy / not working ☹

Program structure	□ very good	□ good	□ poor	/2
Comments within code	□ good comments	□ some comments	□ few/no comments	/1

Functionality:

□ Less than minimal □ Minimal □ Partial □ Complete □ Complete + extensions

Features:

Source code program is:	□ hard coded	□ from file	□ handles multiple programs ☺		/2
Messages (file handling errors):	□ informative	□ minimal	□ none ☹		/2
Translates instructions correctly:	□ all	□ most	□ some	□ none ☹	/3
Translates operands correctly:	□ all	□ most	□ some	□ none ☹	/3
Saves machine code program:	□ yes	□ no	□ optional		/2
Has symbol table:	□ yes	□ no	□ more than one		/2
Passes:	□ two	□ only one ☹			/1
Handles comments correctly:	□ yes	□ no			/2
Handles labels for jumps/variables:	□ yes	□ no			/2

Extensions:

Handles errors in source code:	□ yes	□ no				/2
Messages (source code errors):	□ informative	□ minimal	□ none			/2
On-screen information:	□ informative	□ minimal	□ binary	□ none	□ optional	/2
Extra assembly language programs:	□ yes	□ no				/2

□ Other useful features noted:

Your report and demonstration: **…… out of 20**

□ Clear structure / layout	□ Some structure	□ Poor structure / layout		/ 2
□ Writing style clear & appropriate	□ Style a little confusing	□ Hard to read		/ 2
□ Good length *(600-800 words)*	□ Too long	□ Too short	□ No word count	/ 1
□ Good spelling/grammar	□ A few errors	□ Poor spelling/grammar		/ 2
□ Good description of approach	□ Adequate description	□ Poor description		/ 3
□ Good description of problems	□ Adequate description	□ Poor description		/ 3
□ Good description of solutions	□ Adequate description	□ Poor description		/ 2
□ Good demonstration	□ Adequate demonstration	□ Poor demonstration	□ None	/5

Other comments:

Overall grade (taking account of ..... days lateness) **____ % = ____**

(this grade will be combined with peer review information and converted to the 14% overall value of this assignment within the module)