

## Solutions 6

### Question 1: 7 Marks

- a) Both are fairly low level, unrestricted languages. Both allow the modification of arbitrary memory address. However, C provides more abstraction than assembly in terms of not needing to know the specific instruction set and is hence more portable as C code can easily be recompiled for different platforms. **3 marks**
- b) ENTRY defines where the program will start running when GDB is controlling it. The reset vector defines where the program will start running when the micro is not being controlled by GDB. Typically these values are made the same but they could be different as they are defined differently. **2 marks**
- c) -S: stop after compiling from C to assembly. Don't produce object code or link.  
-Wall: enable all compile warnings. **2 marks**

### Question 2: 6 Marks

- a) We get precise control over where sections are placed in memory. **1**
- b) .text **1**
- c) It is critical that the vectors are placed at a very specific location in memory. **1**
- d) out\_section : { \*(in1) bar.o(in2) } > RAM **3**

### Question 3: 10 Marks

- a) The integer data types defined with a specific size. ie: uint32\_t etc. **1**
- b) The compiler expects main to return int and will throw a warning if it not int **1**.
- c) Main takes no arguments **1**
- d) Treats the number 0x08000100 as a pointer to 16 bits of signed data and accesses the 16 bits of data pointed to by the pointer (aka: at the address). **2**
- e) With the data type it means the data type is a pointer type. By itself: it means dereference the pointer or access the data being pointed to by the pointer **2**
- f) **3**

```
LDR R0, SRC_ADDR
LDR R0, [R0]
ADDS R0, R0, #42
LDR R1, DST_ADDR
STRH R0, [R1]
.align
```

SRC\_ADDR: .word 0x2000FF0

DST\_ADDR: .word 0x2000000