The overall performance for question 1 (basic stuff) is generally satisfactory which shows that the students have been able to grasp the fundamentals of the course. Some specific inaccuracies done by a small fraction of the students are:

a) The important thing about 3-address style is that all the operands and result are read/written from/to memory.
   i) Having instructions and data together in memory is not enough to define Von Neuman arch., that's what differentiates it from Harvard, but not from other architectures. One important part of the architecture is the bus, which becomes the bottleneck.
   ii) The important thing to be mention about registers is that they can be accessed faster than mem.
   iii) Pseudo instructions not necessarily need to be translated into more than one instruction and MUL doesn't need to be a pseudo instruction (depends on the architecture).
   iv) Assembler Directive was confounded rather often with the assembly process. The important part here is that directives are not executed, but tell the assembler how to define initial program state.

b) the main issues in b are that the conditional instructions don't compare the operands but use the status register and that STRB copies the least significant (top, bottom, first and last have been accepted but are inaccurate) byte to memory WITHOUT zeroing the rest of the bytes of the word.

c) The main difference between & and && is that && is lazy, i.e. don't check the second expression if it knows the result from the first -- e.g. 0 && x or 1 || y

Question 2 seems to have been more challenging for students than anticipated but the overall performance has still been good enough and in general consistent with the results in Q1. Some common mistakes from this part are:

a) direct addressing does not need to be faster than others
   range of indirect addressing doesn't need to be limited (we can use a register)
   offset can be EITHER a literal or a register
   & v) good in general. Some students have mixed them up which has been penalized with -1/2 mark (1.5 rather than 2)

b) Some students have referred to the address table as 'stack', this has penalized with -1 mark. A few students have implemented string=text[day] rather than a proper switch. Although this is not a valid implementation of the switch I have accepted it because it shows out-of-the-box thinking and, indeed, uses a mechanism very similar to the address table.

c) This question aimed to check whether students understood methods and stacks. For this reason I've been quite lenient when looking at the code and have even looked at crossed and rough work. I haven't accepted iterative implementations as that was not what students were asked for.

See the attached for feedback from RSN
Section A

General Feedback Comments
The following general comments are suggested to make your reflection and feedback more readable/succinct and viable for a more general succinct meditation on what you could do to enhance your learning and may be adapt your revision methodology.
First, it is important to reflect on the last lecture RN presented to the cohort; at the end of that lecture a comprehensive set of steps and guidance was presented for revision that students have advised me that they utilise. This was derived from methods students have utilised to revise over the years. The sentiments and guidance in this revision was sometimes given in their own words [the students]; and their own reflection on what worked best for different situations. But, may be it is worth noting that sometimes there can be discrepancies between the student’s view of what mark they should have attained and what they actually were awarded. Reflecting on this issue may be it is worth noting and quoting the specific feedback from a student with respect to using the [suggested] self-test (or self-assessment) methodology, they said:

“With regards to the self-assessment questions definitely allowed me to retain and recall large amounts of domain knowledge.
It was especially useful in shorter questions and proved more beneficial than simply re-reading notes in a repetitive manner.”

This is pertinent as without utilising a method like the self-assessment as well as undertaking a number of past exam papers one cannot self-access one’s ability to pass the exam or access what mark one may obtain; and it is even more important when one goes into industry as without a good understanding of your own abilities how can you decide which courses to take, either those presented by the company that employs you or by external courses, and hence how can you evolve your personal skill set. You could also assesses where you feel you are in the The Four stages of Learning (4SoL)): or Do you know what you know?, information on this [4SoL] is on Blackboard 9.

Good companies will encourage you to undertake CPD. Continuing professional development (CPD) or Continuing professional education (CPE) is the means by which people maintain their knowledge and skills related to their professional lives.
One could say that evolving your revision and exam skill at University is, in fact, a form of CPD.

A final point, before getting into the detailed feedback for each question, is to reflect or ask yourself questions such as:
Did I undertaking past exam papers? [enough]
Did I undertaking a past exam paper –timed? [to get used to the time constraint of a real exam]
Did I develop a self-test (or self-assessment)? and finally
Did you utilise the methodologies presented in RN’s exam revision lecture?

One of the recurring points was the advice given on diagrams does not seem to have been noted by many students; hence we repeat it hear: Remember, good – honours grade answer – in the exam – for a question – to maximise marks – should – or you should think of adding A DIAGRAM or a set of diagrams; hence a basic layout of a question answer may be:
1) Textual answer;
2) Diagram supporting answer [or code snippet]; &
3) Full explanation of diagram…
This sort of answer will [may] maximise your marks…

Next, well done to all of you, the answers you gave were some of the best I have seem, and particularly for question 3, which was a question that was well structured and its complexity increased at each stage; but you all seemed to deal with it extremely well; again well done. I do not know what to put this down to; could be your excellent revision [hopefully my advice on that front worked]; or may be one of the lecturers was good; or you studied hard; whatever it was the skills you displayed here will help you in industry.
d) What is the ‘heap’ in a Java Virtual Machine? Your explanation should include an explanation of when it is used and what is stored there. (2 mark)

1.e.

**Bookwork (2 marks):**

The following points should be covered to some degree in the answer:

The heap is an area of permanent memory which is allocated dynamically when needed. Its usual use in the implementation of Java is to store information associated with an instance of an object. The memory is allocated when a `new` is executed. The major content of the memory allocated is the instance variables of the object. However, there is also extra information associated with the class, size and structure of the object.

**ALSO:**

The JVM has a memory manager. This contains a subroutine which is called whenever a ‘new’ is executed. The JVM has an area of store, the heap, where the memory manager allocates memory for newly created objects. This is clearly dynamic, i.e. the heap grows as the program runs.

**2 marks** for an answer that depicts all the salient facts in a sensible way; all facts are correctly delineated and briefly described;

**1 ½ marks** for a right-lines approach;

**1 mark** for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s): Lecture 18: Java Memory Usage.

TOTAL marks (2 marks) [2]

**Marker’s feedback**

**Pedagogic assessment [criterion]:**

The question assesses Lecture 18: Java Memory Usage.

Well done, most of you were able to state what a ‘heap’ in a Java Virtual Machine was [and describe it and you stated how it was used].

The question’s answer should clearly evidence knowledge of required salient facts relating to the function of a ‘heap’ in a Java Virtual Machine.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: area, permanent memory, allocated, dynamically, store information etc., see above template answer.

Main differentiation that must be clearly evidenced in your answer is that: heap is an area of permanent memory which is allocated dynamically – given its usual use in the implementation of Java is to store information associated with an instance of an object.
The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to the heap; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism [how, and what the heap is] plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.

This theory of heap was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.

e) Explain the term ‘Garbage Collection’ and explain why it is important to the implementation of Java. (4 marks)

Application (Critique) (4 marks):

The following points should be covered to some degree in the answer:

i) Programs which use dynamic memory allocation may make use of the memory for a period but then, after a time, use it no longer. In these circumstances, a Garbage Collector tries to identify dynamically allocated memory which is no longer used and returns it to the memory allocation system.

Also, as we are continually creating new objects, we can run out of store. We would like to reclaim store; if it is no longer useful. An important part of the memory manager is a garbage collector. Called when the heap is getting full. Works out what is ‘garbage;’ e.g. un-referenced objects; no references [to it] exist …

ii) In Java, objects are continually allocated and discarded, so GC is very important. If this was not done by the GC the heap would fill up with un-referenced [unused] objects; this is important as the heap [nominally] has a set size (or limit on how much memory it can use) so the GC will stop the heap becoming full and remove unwanted [un-referenced] objects.

4 marks for an answer that depicts all the salient facts in a sensible way; good briefly described and expression is evaluated totally correct;

3 marks for correct answer but not detailed [enough];

2 mark for a right-lines approach;

1 mark for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s): Lecture 18: Java Memory Usage.

TOTAL marks (4 marks) [6]

Marker’s feedback

Pedagogic assessment [criterion]:
The question assesses lecture 18: Java Memory Usage.
Well done, most of you were able to state what function the ‘Garbage Collection’ performed and you stated how it performed its function. The question’s answer should clearly evidence knowledge of required salient facts relating to the function of a ‘Garbage Collection’.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: dynamic memory, allocation, identify, dynamically, allocated, memory etc., see above template answer.

Main differentiation that must be clearly evidenced in your answer is that: Garbage Collector tries to identify dynamically allocated memory which is no longer used and returns it to the memory – given the it works out what is ‘garbage;’ e.g. un-referenced objects; no references [to it] exist.

The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to the ‘Garbage Collection’; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.

This theory of the ‘Garbage Collection’ was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.
f) Operating System consists of basically two main components: the kernel and its libraries. The kernel manages hardware resources and it has a set of managers to manage these basic resources. Name the three managers and briefly describe each. (4 marks)

1.g.
Bookwork & explanation (4 marks):

g) Operating System consists of basically two main components: the kernel and its libraries. The kernel manages hardware resources and it has a set of managers to manage these basic resources. Name the three managers and briefly describe each. (4 marks)

The following points should be covered explicitly in any exam; to gain full marks for your answer:

The three-managers and descriptions are:

1. **Process Manager (CPU):** creates the illusion that **several things** can be going on **at once**.
2. **Memory Manager (main memory):** each **program** has all the **memory** it needs, but can’t access anything else.
3. **Peripheral Managers:** separate manager (**device driver**) for each (**kind of**) **peripheral**.

**4 marks** for an answer that depicts all the salient facts in a sensible way; good briefly described and expression is evaluated totally correct;

**3 marks** for correct answer but not detailed [enough];

**2 mark** for a right-lines approach;

**1 mark** for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s.): Lecture 15: System Software.

TOTAL marks (4 marks) [10]

**Marker’s feedback**

**Pedagogic assessment [criterion]:**
The question assesses lecture 15: System Software.

Well done, most of you were able to name the three [main] kernel managers. The question’s answer should clearly evidence knowledge of required salient facts relating to the function the three [main] kernel managers performed.

In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: **Process Manager, several things, running, at once**; **Memory Manager, each program, [allocated] memory [all mem. it needs]**; **Peripheral Managers, device driver, peripheral** etc., see above template answer.
Main differentiation that must be clearly evidenced in your answer is that: you must correctly names [three] managers – given the managers names; a brief description [as per the template answer] should be given of each – with the appropriate keywords used for the description of each manager – and putting these keywords in context, for the task they perform.

The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to the three different manages [in the kernel]; this was not done explicitly in some of the answers given. If the answers did not detail the three mechanism plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.

This theory of the three kernel managers was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.
Section C

3.

a) In your lectures and exercise classes you were introduced to code, in the form of a loop, that can check a table of peripherals to see which caused an interrupt; which would be something like that presented in figure 3.a; except that the code in the figure is out of order! Reorder the code into the correct order in your answer so it enables an interrupt handler to deal with several peripherals. Note: the data structure the code uses is given above the out-of-order code in figure 3.a.

(5 marks)

<table>
<thead>
<tr>
<th>Addresses</th>
<th>Labels</th>
<th>Instruction</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000088</td>
<td>Address_0_StatusRegister</td>
<td>DEFW</td>
<td>0x40000004</td>
</tr>
<tr>
<td>0x0000008C</td>
<td>peripheral_0_TestPattern</td>
<td>DEFW</td>
<td>0x00000080</td>
</tr>
<tr>
<td>0x00000090</td>
<td>Address_1_StatusRegister</td>
<td>DEFW</td>
<td>0x40000008</td>
</tr>
<tr>
<td>0x00000094</td>
<td>peripheral_1_TestPattern</td>
<td>DEFW</td>
<td>0x00000100</td>
</tr>
</tbody>
</table>

(Data structure)

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEQ</td>
<td>loop</td>
<td></td>
</tr>
<tr>
<td>TST</td>
<td>R1, R2</td>
<td></td>
</tr>
<tr>
<td>LDR</td>
<td>R2, [R0], #4</td>
<td></td>
</tr>
<tr>
<td>ADR</td>
<td>R0, table</td>
<td></td>
</tr>
<tr>
<td>LDR</td>
<td>R1, [R1]</td>
<td></td>
</tr>
<tr>
<td>LDR</td>
<td>R1, [R0], #4</td>
<td></td>
</tr>
</tbody>
</table>

(Code)

Question figure 3.a. Data structure and code; showing out-of-order code.
3.a.
Application (example re-coding) (4 marks):
The following points should be covered to some degree in the answer:

The typical ordered code is:
1: ADR R0, table ;
2: loop
3: LDR R1, [R0], #4 ;
4: LDR R1, [R1] ;
5: LDR R2, [R0], #4 ;
6: TST R1, R2 ;
7: BEQ loop ;

From the previously out-of-order code:
7: BEQ loop ;
6: TST R1, R2 ;
5: LDR R2, [R0], #4 ;
1: ADR R0, table ;
4: LDR R1, [R1] ;
3: LDR R1, [R0], #4 ;
2: loop

4 marks for majority of above; e.g. for an answer that orders all lines correctly,
3 marks for correct answer but not all correct order [enough],
2 mark for some in order & for a right-lines approach,
1 marks for some basic understanding (or attempt).

Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 14: Input/Output (2).

TOTAL marks (4 marks) [4]

Marker’s feedback
Pedagogic assessment [criterion]:
The question assesses lecture Lecture 14: Input/Output (2).

Well done, most of you were able to re-order the code into the correct order.
The question’s answer should clearly evidence knowledge of required code order relating to the function or in this case [again] the order of code; that then performs the correct function [e.g. check a table of peripherals] – only if it is placed in the correct order.
In the answer [some of] the appropriate and correct ordering is the important feature of this question.
Main differentiation that must be clearly evidenced in your answer is that: you comprehend the correct order of code – by re-ordering the code in to the appropriate and correct order.
The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to the correct code order; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism [order] plainly as stated in the
above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.

This theory of check a table of peripherals and the code to undertake this function was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.
b) Given the code, in figure 3.b. enables an ARM CPU to poll a simple device.

The table, in figure 3.b, shows a code snippet to poll a simple device. Copy the table (3.b.) into your answer book (with a comments column) and then describe in full what happens when the ARM program is obeyed. In the table clearly describe the movement of information (both values and instructions) between memory, registers and the CPU in the comments column, and how the values in the registers R0, R1, and memory change, at each step. (5 mark)

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
<th>Detailed comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop</td>
<td>ADR</td>
<td>R1, Status_Reg</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TST</td>
<td>R0, #0x80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BEQ</td>
<td>loop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ADR</td>
<td>R1, Data_Reg</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVC</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Status_Reg</td>
<td>DEFW</td>
<td>0x80</td>
<td></td>
</tr>
<tr>
<td>Data_Reg</td>
<td>DEFW</td>
<td>0x72</td>
<td></td>
</tr>
</tbody>
</table>

Question figure 3.b. A table; requiring adding comprehensive comments.
3. b. Application (example details) (5 marks):
The following points should be covered to some degree in the answer:

The typical fully commented ordered code is:

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
<th>Detailed comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop</td>
<td>ADR</td>
<td>R1, Status_Reg</td>
<td>; R1 points to status reg.; load address of ‘Status_Reg’ in to R1</td>
</tr>
<tr>
<td></td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td>; read status reg. [of device]; using R1 to point at address of ‘Status_Reg’</td>
</tr>
<tr>
<td></td>
<td>TST</td>
<td>R0, #0x80</td>
<td>; test ready bit (bit 7); test [or AND] R0 with immediate value 0x80</td>
</tr>
<tr>
<td></td>
<td>BEQ</td>
<td>loop</td>
<td>; if not ready, try again; branch if bit 7 = 0 (zero); or if equal zero, zero bit set in status reg.</td>
</tr>
<tr>
<td></td>
<td>ADR</td>
<td>R1, Data_Reg</td>
<td>; R1 points to data reg. [of device] ; load address of ‘Data_Reg’ in to R1</td>
</tr>
<tr>
<td></td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td>; read data [from data register]; using R1 to point at address of ‘Data_Reg’</td>
</tr>
<tr>
<td></td>
<td>SVC</td>
<td>2</td>
<td>; end [exit or stop] program</td>
</tr>
<tr>
<td>Status_Reg</td>
<td>DFW</td>
<td>0x80</td>
<td>; data definition, define word</td>
</tr>
<tr>
<td>Data_Reg</td>
<td>DFW</td>
<td>0x72</td>
<td>; data definition, define word</td>
</tr>
</tbody>
</table>

5 marks for majority of above; e.g. for an answer that orders all lines correctly (2 marks for a detailed and concise description.),
3 marks for correct answer but not concise [enough],
2 mark for some information & for a right-lines approach,
1 marks for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s).: Lecture 14: Input/Output (2).

TOTAL marks (5 marks) [10]

Marker’s feedback

Pedagogic assessment [criterion]:
The question assesses Lecture 14: Input/Output (2).

Well done, most of you were able to describe the movement of information.
The question’s answer should clearly evidence knowledge of required salient facts [on a line by line basis] relating to the function of describing the movement of information.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example: Load, address, register, status, register, etc., see above template answer.
Main differentiation that must be clearly evidenced in your answer is that: a full description of the movement of information must include: all the appropriate keywords described in the correct context.
The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to describing the movement of information; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.

This theory relating to describing the movement of information was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.
c) Given, in figure 3.c, that the program polls a simple device. In the context of data exchange between CPU and peripherals: describe in detail exactly what happens when the following ARM program is obeyed.

clearly describe the movement of information (both numbers and instructions) between the peripheral and the CPU, and how the values in the registers R0, and R1 change at each step. Assume that the program starts at memory location 0000 and the Status_Reg contains 0x80 while the Data_Reg contains 0x84 or ascii character 'H.' R0 and R1 should be the values after each [row] instruction has been executed. (4 marks)

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
<th>R0</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>loop</td>
<td>ADR R1, Status_Reg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0008</td>
<td>TST</td>
<td>R0, #0x80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000C</td>
<td>BEQ</td>
<td>loop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>ADR</td>
<td>R1, Data_Reg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0014</td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0018</td>
<td>SVC</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>001C</td>
<td>Status_Reg</td>
<td>DEFW 0x80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>Data_Reg</td>
<td>DEFW 0x72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question figure 3.c. A table; of program polls a simple device.
3.c.
Application (example re-coding) (4 marks):
The following points should be covered to some degree in the answer:

Registers R0 and R0 are sequentially loaded with the following data:

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
<th>R0</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>loop</td>
<td>ADR</td>
<td>-</td>
<td>0x001C</td>
</tr>
<tr>
<td>0x0004</td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td>0x80</td>
<td>0x001C</td>
</tr>
<tr>
<td>0x0008</td>
<td>TST</td>
<td>R0, #0x80</td>
<td>0x80</td>
<td>0x001C</td>
</tr>
<tr>
<td>0x000C</td>
<td>BEQ</td>
<td>loop</td>
<td>0x80</td>
<td>0x001C</td>
</tr>
<tr>
<td>0x0010</td>
<td>ADR</td>
<td>R1, Data_Reg</td>
<td>0x80</td>
<td>0x0020</td>
</tr>
<tr>
<td>0x0014</td>
<td>LDRB</td>
<td>R0, [R1]</td>
<td>0x72</td>
<td>0x0020</td>
</tr>
<tr>
<td>0x0018</td>
<td>SVC</td>
<td>2</td>
<td>0x72</td>
<td>0x0020</td>
</tr>
<tr>
<td>0x001C</td>
<td>Status_Reg</td>
<td>DEFW</td>
<td>0x80</td>
<td>-</td>
</tr>
<tr>
<td>0x0020</td>
<td>Data_Reg</td>
<td>DEFW</td>
<td>0x72</td>
<td>-</td>
</tr>
</tbody>
</table>

4 marks for majority of above; e.g. for an answer that orders all lines correctly (2 marks for a detailed and concise description.),
3 marks for correct answer but not concise [enough],
2 mark for some information & for a right-lines approach,
1 marks for some basic understanding (or attempt).
Reference Learning Resources, Background Reading, and Lecture itself for detailed information; Lecture(s) No(s): Lecture 13: Input/Output (1).
TOTAL marks (4 marks) [14]

Marker’s feedback
Pedagogic assessment [criterion]:
The question assesses Lecture 14: Input/Output (2).

Well done, most of you were able to state how many times the loop ran and the appropriate values in the registers.
The question’s answer should clearly evidence knowledge of required salient facts relating to how many times the loop ran and the appropriate values in the registers.
Main differentiation that must be clearly evidenced in your answer is that: must state correctly the appropriate [and correct] values in the registers at the end of the loop cycle.
The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to how many times the loop ran and the appropriate [and correct] values in the registers at the end of the execution of each instruction; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.
Most marks were lost as not enough attention to detail was paid when answering the question; this ranged from not clearly and precisely aligning each instruction [address in memory] with explicit R0 and R1 values for each instruction; also badly written answers lead [the markers and examiners] to having to resort to either guessing what the answer was
meant to imply or worst as the answer was not written in clear legible hand writing (and good English and grammar) it was extremely hard to try to decipher the answer; a number of students (however) did tabulate: clear, precise, readable answers, and the highest mark were given to answers that aligned to the template answer above; whereas un-readable or undecipherable answers could not be marked or awarded reasonable marks.

This theory behind assigning the values in the registers at the end of each instruction was covered in the lecture series; in labs and exercise classes; and may have been delivered in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.

3.d. Application (example) (Critique) (6 marks):

d) Given the code in 3.b. is now defined as ‘Status_Reg DEF 0x00’:

i) What would you expect to happen; when the program sequences through lines 0x0000 to 0x000C?
In your answer also state:

ii) what decision has the loop enabled the processor to make with respect to polling a simple device? And finally:

iii) Is the branch (BEQ) activated; and if so what caused the branch?

In your answers to i) to iii) please be explicit and comprehensive when you write the answers to each subsection. (6 marks)
Marker’s feedback

Pedagogic assessment [criterion]:
The question assesses Lecture 14: Input/Output (2) objective 1: Distinguish between ARM’s interrupts [interrupt handler].

Well done, most of you were able to state “What would you expect to happen next” and answering parts (b) and (c) concisely.
The question’s answer should clearly evidence knowledge of required salient facts relating to “What would you expect to happen next”.
In the answer [some of] the following terminology (keywords and naming conventions) should be utilised in context; for example this includes all the key words underlined in the example answer above.
Main differentiation that must be clearly evidenced in your answer is that: “What would you expect to happen next” is clearly, concisely and explicitly stated – and (b) & (c) are full addressed.
The question’s answer should clearly evidence knowledge of required salient facts relating to the issues aligned to “What would you expect to happen next”; this was not done explicitly in some of the answers given. If the answers did not detail the mechanism plainly as stated in the above (and in the Example answer); e.g. if the answer did not give evidence of knowledge of required salient facts, full marks were not awarded.
This theory of polling a simple device was covered in the lecture series in three different audio visual media: 1) the actual live lecture; 2) the audio recording of the live lecture; and 3) the real time video of the lecture.

END OF EXAMINATION