QUESTION 1.

3 **(a)** The 245th page frame from the start of memory
// the 245th page frame from some base address

(b) Flash memory // magnetic disk // hard drive

(c) **(i)** Time of entry (NOT time in memory) [1]

(ii)

Page	Presence Flag	Page frame address	Additional data
4	1	542	12:07:34:49

[1 +1 + 1]

(iii) Number of times the page has been accessed [1]

(iv)

Page	Presence Flag	Page frame address	Additional data
3	1	132	0

[1 +1 + 1]

Accept only zero for 'additional data'

(d) For example:
Longest resident: page in for lengthy period of time may be being accessed often [1]
… so not a good candidate for being removed [1]

Least used: a page just entered has a low least used value … [1]
so likely to be a candidate for immediately being swapped out [1]

QUESTION 2.

3 **(a)** The page is present in memory
Loaded at / stored /present in page frame 542 // its memory address is 542

(b) **(i)** Next instruction is first instruction in Page 6
Page 6 is not present in memory [1]
Instruction can only be executed if present in memory [1]
Program cannot continue until Page 6 is loaded [1]

[Max 2]

(ii) When there is an attempt to load an instruction for a page not in memory [1]
A page fault occurs // Page 5 finishes … [1]
this generates an interrupt [1]
ISR code is executed [1]
Causes the OS to load page 6 into memory [1]

[Max 3]

(c) **(i)** Time of entry (NOT time in memory) [1]

(ii)

Page	Presence Flag	Page frame address	Additional data
6	1	221	12:07:34:49

[1 + 1 + 1]

(iii) When the procedure call is made – Page 1 is swapped out and Page 3 is swapped in [1]
At the end of the procedure call – Page 3 is swapped out and Page 1 is swapped in [1]
Page 1/3 is always in memory shortest amount of time [1]
The entire sequence is repeated for every iteration [1]

[Max 3]

(iv) Thrashing // continually swapping pages [1]

QUESTION 3.

3 **(a)** The 245th page frame from the start of memory
// the 245th page frame from some base address

(b) Flash memory // magnetic disk // hard drive

(c) **(i)** Time of entry (NOT time in memory) [1]

(ii)

Page	Presence Flag	Page frame address	Additional data
4	1	542	12:07:34:49

[1 +1 + 1]

(iii) Number of times the page has been accessed [1]

(iv)

Page	Presence Flag	Page frame address	Additional data
3	1	132	0

[1 +1 + 1]

Accept only zero for 'additional data'

(d) For example:
Longest resident: page in for lengthy period of time may be being accessed often [1]
… so not a good candidate for being removed [1]

Least used: a page just entered has a low least used value … [1]
so likely to be a candidate for immediately being swapped out [1]

QUESTION 4.

Question	Answer	
6(a)(i)	Control system	
6(a)(ii)	Use of actuators means that the system is controlling	
6(b)	System wastes processor time checking for values that are not changing 1 Some sensor input needs to be acted upon immediately 1	**2**
6(c)(i)	Interrupts need to be disabled so that the process of dealing with an interrupt is itself not interrupted	**1**
6(c)(ii)	After handling the interrupt interrupts need to be enabled so that further interrupts can be dealt with	**1**
6(c)(iii)	Content of registers 1 Placed on stack 1	**2**
6(c)(iv)	Changing sensor value dealt with as soon as it happens 1 Processor needs to check sensor only when an interrupt occurs 1	**2**
6(c)(v)	AND #B0000001000000000 // AND #&0200 // AND #512 Op code 1 Operand 1	**2**

QUESTION 5.

Question	Answer	Marks
5(a)(i)	1 mark per bullet point: • Running process is halted • Process moves to blocked state	**2**

Question	Answer	
5(a)(ii)	1 mark per bullet point max 2: • Running process is halted // another process has use of the processor • Process moves to ready state • … Until next time slice allocated	
5(b)(i)	1 mark per bullet point: • Current process no longer running // processor is available • Process was at the head of the ready queue / / process has highest priority	**2**
5(b)(ii)	1 mark per bullet point: • The only • Required resource becomes available // event is complete	**2**
5(c)	1 mark per bullet point to max 3: • to allow multiprogramming • to give each process a fair share of the CPU time • to allow all processes to complete in a reasonable amount of time • to allow highest priority jobs to be executed first • to keep the CPU busy all the time • to service the largest possible number of jobs in a given amount of time • to minimize the amount of time users must wait for their results • to maximise the use of peripherals	**3**

QUESTION 6.

	• Dictionary	
6(a)	**1 mark** per bullet point Page: • Virtual Memory is divided into blocks of a fixed size Page frame: • the main memory is divided into page frames of the same size as a page Page table: • the Page (Map) table shows the mapping of pages to page frames	**3**

Question	Answer	
6(b)	**1 mark** per bullet point to **max 3** • To allow multiprogramming / multitasking to take place • To ensure fair usage of the processor • To ensure fair usage of peripherals • To ensure fair usage of memory • To ensure higher priority tasks are executed sooner • To ensure all processes have the opportunity to finish	
6(c)	A **signal** from a software **source** or hardware device seeking the **attention of the processer**	**1**
6(d)	**1 mark** per bullet point in the order given • JOB32 • JOB42 • JOB42	**3**

QUESTION 7.

Question	Answer	
4(a)	1 mark per bullet point to **max 2** • Disk / secondary storage is used to extend the RAM / memory available • … so CPU can access more memory space than available RAM • Only part of program / data in use needs to be in RAM • Data is swapped between RAM and disk	
4(b)(i)	1 mark per bullet point to **max 4** • Divide memory / RAM into frames • Divide virtual memory into blocks of same size called pages • Frames / pages are a fixed size • Set up a page table to translate logical to physical addresses • Keep track of all free frames • Swap pages in memory with new pages from disk when needed	**4**
4(b)(ii)	First-in-first-out // least-recently-used page // least-used-page	**1**
4(b)(iii)	1 mark per bullet point to **max 2** • Pages are required back in RAM as soon as they are moved to disk • There is **continuous** swapping (of the same pages) • No useful processing happens // deadlock • … (because) pages that are in RAM and on disk are inter-dependent • … (nearly) all processing time is used for swapping pages	**2**