BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT

COMPUTER & NETWORK TECHNOLOGY

Examiners' Report

General comments on candidates' performance

The average mark and overall pass rate were low.

Candidates are advised to provide clear, structured answers and submit these answers in the correct answer booklet.

A1

a) Desktop and workstation computers have changed immensely over the last decade. Discuss how the general-purpose desktop computer has changed over the last few years in terms of hardware, software, and applications. Your answer should include a reference to the way in which the performance has changed.

(20 marks)

b) Over the years, the clock rate of computers has increased, rising from 1 MHz in the 1970s to 4 GHz today. However, over the last few years, clock speeds have not increased greatly. Briefly explain why the clock speed of modern processors is limited, and state what engineers are doing to overcome this limitation.

(5 marks)

c) You want to buy 100 new computers for your organization. You shortlist two contenders; machine *A* and machine *B*. These computers each run four tasks; *P*, *Q*, *R*, and *S*. The table below gives the performance data; for example, task *P* runs on machine *A* in 20 seconds. Task *P* occupies 10% of the load of the computer; that is, the task occupies 10% of the jobs/programs being run on computer A.

Use the data to say, with a brief reason, which computer is the better.

Task	Machine A		Machine B	
	Time	Load	Time	Load
P	20s	10%	30s	10%
Q	100s	5%	60s	10%
R	5s	70%	10s	60%
S	10s	15%	15s	20%

(5 marks)

Answer pointers

a)

This was an open-ended question. The purpose of the question was to test whether the candidate had an understanding of how computer systems are developing and what constitutes a computer system.

The changes in a computer system's hardware have largely been in three areas: processor, memory, I/O

Processor: For decades, computing has been driven by Moore's Law; a rule-of-thumb that states (approximately) computer power doubles every 18 months. Consequently, processors have been getting faster – partially due to manufacturing technology and partially due to internal design (architecture and organization such as new instructions). The effect of this progress has been to increase the range of applications that a computer can run. Four decades ago, computers struggled with text processing. High-speed technology now means that they can perform real-time operations on video data and implement operations such as virtual reality and immersive games.

Memory: Memory density (the number of bits that can be stored in a given volume) has increased immensely. This means that programs can be far larger and therefore process more data and perform more operations. In the past, a large program could not be fitted in main memory and parts of it had to be swapped between main memory and disk – a slow process. Today, direct access (main store) memory sizes of up to 64GB are not uncommon.

The advance in secondary storage with disk drives increasing from 5 MB to 10TB over about four decades means that very large data sets can be stored. Note that the speed of hard drives has increased very little over the decades. This is because they are mechanical devices and it is impossible to spin disks more rapidly and to moves heads across the surface faster. However, a recent innovation is the use of SSD (solid state disks) that employ semiconductor flash memory. These behave like conventional hard disks but are faster, more reliable, and consume less power. Already, all high-end laptops use SSDs and many high-performance PCs have one SSD to hold the operating system.
Modern storage technology (i.e., advances in speed and storage density) means that very large data structures can be handled … this is important in real-time image processing applications and in scientific applications involving modelling physical systems (i.e., simulation).

Input/Output: Display technology has rapidly advanced. Originally, computers had text-based displays which were replaced by windows graphical environments. Today, many displays are high resolution, offering pixel densities so great that the eye cannot detect any further increases in resolution. Modern displays are sometimes touch sensitive (especially in tablets and personal computers) which means that the mouse and other pointing devices are not necessary. The increase in pixel resolution has led to the development of fast video display cards. These display cards can also perform some of the computation that would normally be done by the CPU.

Interface technology has improved immensely since the early days of the serial and parallel interface. USB, WiFi, and Bluetooth have all combined to make it very easy to connect a wide range of I/O devices to a computer without having to have any knowledge of the physical interface or device drivers. Almost always (today) a peripheral establishes a connection of a computer automatically and invisibly.

Examiner's comments

The evidence shows that many candidates wrote general knowledge about the development of computers starting from vacuum tube based machines and did not answer the question correctly. Many candidates confused "software" and "applications" – better wording would be to ask about changes in "application areas". Request to "include a reference to the way in which the performance has changed" was not addressed sufficiently.

b)

The power consumed by a processor is proportional to its clock frequency. As the clock frequency has increased, the power consumption has also increased which means that it is necessary to remove excessive heat. Now we have reached the limits of conventional cooling (heat sink and fan). Water cooling is expensive and complex and that allows higher clock rates – but we are close to the maximum clock speeds.
Engineers have attempted to deal with the problem by using multicore technology; that is, the chip contains several CPUs running in parallel. This allows more computation without increasing clock rates.

Examiner's comments
The evidence shows that the majority of candidates repeated/re-phrased the question itself and partially repeated what they said in A1a. Request to "explain why the clock speed of modern processors is limited" was addressed inadequately.

c)

We can calculate how much time each computer requires in total by multiplying the amount of time taken by each a job (task) by the fraction of the total number of jobs it takes, and then summing over all jobs. For machine A we have:

Time_A = 20 x 0.10 + 100 x 0.05 + 5 x 0.70 + 10 x 0.15 = 2 + 5 + 3.5 + 1.5 = 12s

Time_B = 30 x 0.1 + 60 x 0.1 + 10 x 0.60 + 15 x 0.20 = 3 + 6 + 6 + 3 = 18s

Therefore, computer A is better than computer B.

Examiner's comments
The evidence shows that the majority of candidates were quick to select "machine B" without satisfactory explanations.

A2

In assembly language, an addressing mode specifies how an operand is accessed by an instruction.

a) There are three fundamental addressing modes used by processors. These addressing modes are called:

   i) Immediate addressing (or literal addressing)
   ii) Memory direct addressing (or absolute addressing)
   iii) Register indirect addressing (or indexed or pointer-based addressing).

   Briefly explain what each of these addressing modes means and provide an example of its application.

   (9 marks)

b) Write an assembly program (using an assembler of your choice) to search a sequence of 64 integers stored in memory, starting at location 100, for the occurrence of 42. If 42 is found, location 200 should be loaded with 1. If 42 is not found it should be loaded with 0.

   Marks will be awarded for your design and documentation (comments) of the program. All instructions you use should be defined. Students may make up (invent)

their own assembly language operations as long as they approximate to real-world operations.

(21 marks)

Answer Pointers

a)

Immediate addressing: The operand is a literal (numeric) value that is part of the instruction. It is used to load a constant into a register. For example, MOVE #25,D0 loads the number 25 into register D0.

Memory addressing uses an operand in the instruction that is the address of data in memory. For example, MOVE Temp4,D3 means copy the contents of memory location Temp4 into register D3. Memory addressing is used to access variables (values that change) in memory.

Register indirect addressing specifies a register in the instruction. This register contains a pointer to the actual operand in memory. The instruction MOVE [D0],D2 means copy the contents of memory pointed at by register D0 to register D2. This addressing mode is used to access tables, vectors, lists and other data structures, because, if the data in D0 (the pointer register) is changed, a different memory location is accessed.

Examiner's comments

Only a few candidates attempted this question and those who did showed a reasonably good understanding of addressing modes. Some applied different notations in the examples (as compared with the model answer) but marks were not reduced because of it.

b)

Designing the program: Here's some pseudocode. Students will be given credit for designing the program before coding it.

```
Set the flag in memory to fail (i.e., 0)
Point to memory location 100 (the starting point)
Repeat
   Read a location
   Compare its contents with 42
   If contents = 42 then set flag to success and exit
   Else increment pointer
   If pointer = 100 + 64 Exit
End Repeat
```

Here's a possible example of the code in RISC style code).

```
    LDR r1,#0     ;set flag to fail (42 not found)
    LDR r3,#200   ;point to flag location in memory… r3 is a pointer
    LDR r1,[r3]   ;set flag in memory to fail
    LDR r0,#100   ;load register r0 with 100 to point to data in memory
Rpt LDR r1,[r0]   ;load r1 with value pointed at by r0
    CMP r1,#42    ;is this the value we want?
    BNE inc       ;if not then continue round the loop
    MOV r0,#1     ;put 1 in r0 (the success flag)
    LDR r2,[r3]   ;save the flag in memory
    B   Ext       ;and exit
inc ADD r0,r0,#1  ;increment memory pointer to next value
    CMP r0,#164   ;check for end of loop
    BNE Rpt       ;Repeat
Ext
```

Note marks awarded for use if pointer based addressing and ability to construct a loop. Here's another example in CISC style code

```
    MOVE #0,200   ;set pointer to fail
    MOVE #100,A0  ;point to data
Rpt  MOVE (A0),D0  ;read a number
    CMP  #42,D0   ;is it 42?
    BNE  inc      ;if not continue round the loop
    MOV  #1,200   ;set the flag
    BRA  Ext      ;and exit
inc  ADD  #1,A0    ;increment pointer
    CMP  #164,A0  ;check for end of loop
    BNE  Rpt
Ext
```

Examiner's comments

Only a few candidates attempted this question but those who did showed a reasonably good understanding of assembly language. The evidence shows that some provided C-like code (which was treated as a pseudo-code) but no assembly code at all which was an unsuitable answer.

A3

a) Convert the decimal number 123 into the following bases:

   i) 2 (binary) in 8 bits
   ii) 16 (hexadecimal) in 4 digits

(4 marks)

b) Why is two's complement arithmetic so widely used by digital computers?

(4 marks)

c) What is the answer to the following binary addition operation? Assume that the numbers are in 8-bit signed two's complement format.

```
 00011101
+11110011
```

(4 marks)

d) Convert the two binary values in part c) and your result into decimal form and comment on the result.

(6 marks)

e) Carefully explain the fundamental differences between binary integers and binary floating-point numbers. What are the relative advantages and disadvantages of each form of number representation?

(12 marks)

Answer Pointers

a)

Part i

123 = 64 + 59 = 64 + 32 + 27 = 64 + 32 + 16 + 11 = 64 + 32 + 16 + 8 + 3 = 64 + 32 + 16 + 8 + 2 + 1
That is, 1111011
However, in 8 bits, this is 01111011

Part ii
We can convert the binary to hex; that is, 0111 1011 = 7B
In for digits this is 007B

Examiner's comments

Many candidates attempted this question and the majority answered it correctly. The evidence shows that some provided binary and/or hex tables, but no structured answer was produced.

b)

Two's complement notation provides only one method of representing negative values. It has several major advantages. First, it is easy to convert a positive value into its two's complement representation by simply inverting all the bits and adding 1. Thus, in four bits +3 = 0011. To get -3 we invert the bits to get 1100 and then add 1 to get 1101.

Two's complement notation provides only one unique value for zero; that is, 000..00. Sone negative representations have two values for zero … which is less convenient.

The sign of a two's complement number is 0 for positive or zero and 1 for negative. This makes testing the sign of a number very easy.

To add or subtract two's complement numbers you use an adder. You don't need a separate subtract or for addition. This simplifies logic design.

Examiner's comments

The evidence shows that only a few candidates were able to answer this question effectively. Most candidates described principles of using 0s and 1s in the computer and not a structured answer.

c)

The sum is 100010000

Examiner's comments

Most candidates attempted this question and the majority answered correctly. The evidence shows that some provided binary tables but did not provide a structured answer.

d)

Students may perform this operation assuming signed or unsigned arithmetic

SIGNED ARITHMETIC
00011101 = +29
11110011 is a negative value = -00001101 = -13.

The sum is 00010000 = +16. (the leading 1 is discarded in two's complement addition)
If we perform the addition +29 +(-13) we get 16 which is the correct value.

UNSIGNED ARITHMETIC
00011101 = +29
11110011 = 243.

The sum is 100010000 = 272

If we perform the addition 29+243 we get 272.
However this is incorrect and is outside the range 0 to 255 allowed by 8 bits in unsigned arithmetic. The leading bit is a carry out.

Examiner's comments
Most candidates attempted this question but only a few answered correctly. Credit was given for the partial answers (e.g. where values such as "29", "243" were given). The evidence shows that some candidates provided binary tables or general discussions, but no structured answer was provided.

e)

Binary integers use strings of 1s and 0s to represent whole numbers (i.e., numbers without fractional parts such as 2, 7, 125, -631). An n-bit binary integer can represent unsigned values in the range 0 to 2n-1 (or signed values in the range -2n-1 to + 2n-1-1). Binary integers are used in computer arithmetic; for example, financial calculations or the calculations of array subscripts or index values; that is, wherever we would use "whole numbers" in everyday life.

Integer arithmetic is accurate. Adding, subtracting, or multiplying integers always yields a correct result (unless the answer is larger than the available number of bits used to represent it).

Unfortunately, integers are not well suited to scientific and engineering calculators involving very large of very small numbers. For this, we use floating point; for example, -1.234 x 1025 is a decimal floating point number. It has a sign, a mantissa (in this case 1.234) and an exponent (here it's 25). The mantissa provides the precision of the number and the exponent its magnitude.

It would be very difficult to perform engineering calculations using integers. Floating-point makes it easier. However, special hardware is needed to implement floating point operations and floating point operations are slower than integer operations.

A very significant difference between integer and floating point operations is that floating point calculations do not yield accurate results. Multiplying X and Y can yield an error. This error is due to the way in which the mantissa is scaled during operations. Consequently, those who use floating point arithmetic have to be aware of its limitations.

Examiner's comments
The evidence shows that most candidates who attempted this question provided extended versions of what they did in A3b, described principles of using 0s and 1s in the computer.

A4

The stack, a data structure, is also called a last-in-first-out (LIFO) queue and is used in many areas of computing. In particular, it is used in the implementation of interrupt structures and subroutine call and return mechanisms.

a) Describe the structure of a stack and explain how it is implemented and maintained in a computer. Your answer should also include a description of the two basic operations that can be applied to a stack.

(10 marks)

b) Briefly explain how the stack is used to implement a subroutine call and return mechanism.

(10 marks)

c) What is an interrupt and what is its role in input/output operations? Show, with the aid of a diagram, how the computer implements an interrupt mechanism.

(10 marks)

Answer Pointers

a)

A stack is a data structure in the form of a queue. Unlike the conventional queue that has two ends, a stack has only one end called top of stack. In a stack, an item is said to be pushed on the top of the stack when it enters and pulled off the stack when it leaves.

Because items enter a stack and leave a stack in reverse order, it is called a last in first out queue.

Although a stack can be implemented in logic with shift registers, computers implement it in memory with a stack pointer that points at the current top of stack. If the stack pointer is SP, then Push X onto the stack is implemented by

SP = SP – 1 (decrement stack pointer to point to the next free item on stack)
M[SP] = X (push X onto the stack by storing it in memory at the new stack pointe value)

An item is removed from stack in the reverse operation; that is,

X = M[SP] (read the item pointed at the stack pointer at the top of stack and copy it to X
SP = SP + 1 (increment the stack pointer to point at the next item down).

NOTE … there are several stack implementations. This description corresponds to the common case of a stack that grows towards lower addresses.

Examiner's comments

Many candidates attempted this question and the majority answered correctly. The evidence shows that some candidates provided general discussions and not structured answers.

b)

A subroutine is a piece of code that can be called from anywhere in a program, executed and an instruction made to the instruction immediately after the calling point.

Suppose an instruction at address N is BSR Y (call the subroutine that begins at address Y) is called. The processor first pushes the value of N+1 on the stack (the return address), the program counter is then loaded with Y and the subroutine executed.

When the subroutine has been executed, an RTS (return from subroutine) instruction is executed and the top item on the stack is copied to the program counter. This is, of course, the return address.

This mechanism ensures that a subroutine can be called and the calling code returned to.
However, the real value of the stack comes from its support for nested subroutines. If a subroutine calls another subroutine, the second subroutine's return address is now put on the top of the stack. When a return is made, it made correctly to the second subroutine.

This mechanism allows subroutines to be nested to any depth and even recursive operations where a subroutine calls itself. This is because the return addresses are accessed in the reverse order in which they were called.

Examiner's comments

Only a small number of candidates attempted this question. The evidence shows that some provided general discussions, but no structured answer was provided.

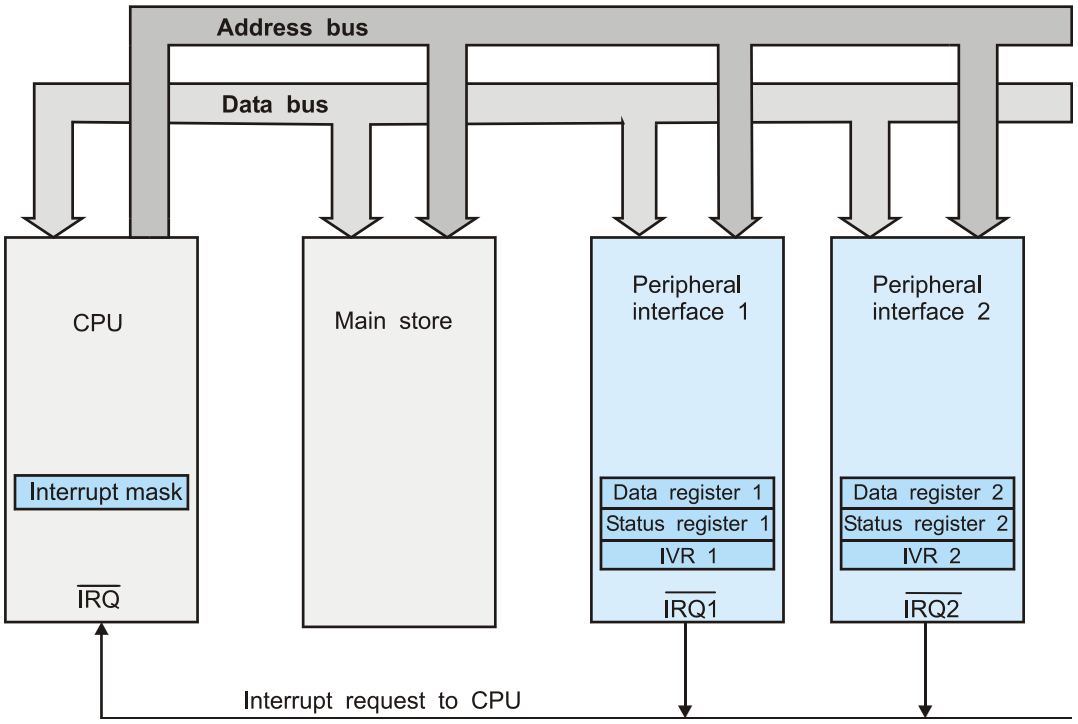
c)

An interrupt or exception is an event that changes the normal sequence of computer operations. Interrupts may be due to errors (divide by zero, non-existent memory access) or requests for attention by I/O devices.

Computers implement interrupt-driven operations in various ways. However, the basic principles are similar in all machines.

In a computer using interrupt-driven input/output, the computer performs tasks normally and does not carry out any explicit I/O activity as part of a program until an I/O device signals that it is ready to take part in a data exchange. That is, the I/O device initiates the I/O transaction; this is the key to interrupt-driven I/O.

The diagram below describes a simple I/O structure. The essential part is a common interrupt request line, IRQ, that connects all peripherals to the computer. When any peripheral requires attention, it asserts (drives) the IRQ line and the CPU detects this event.

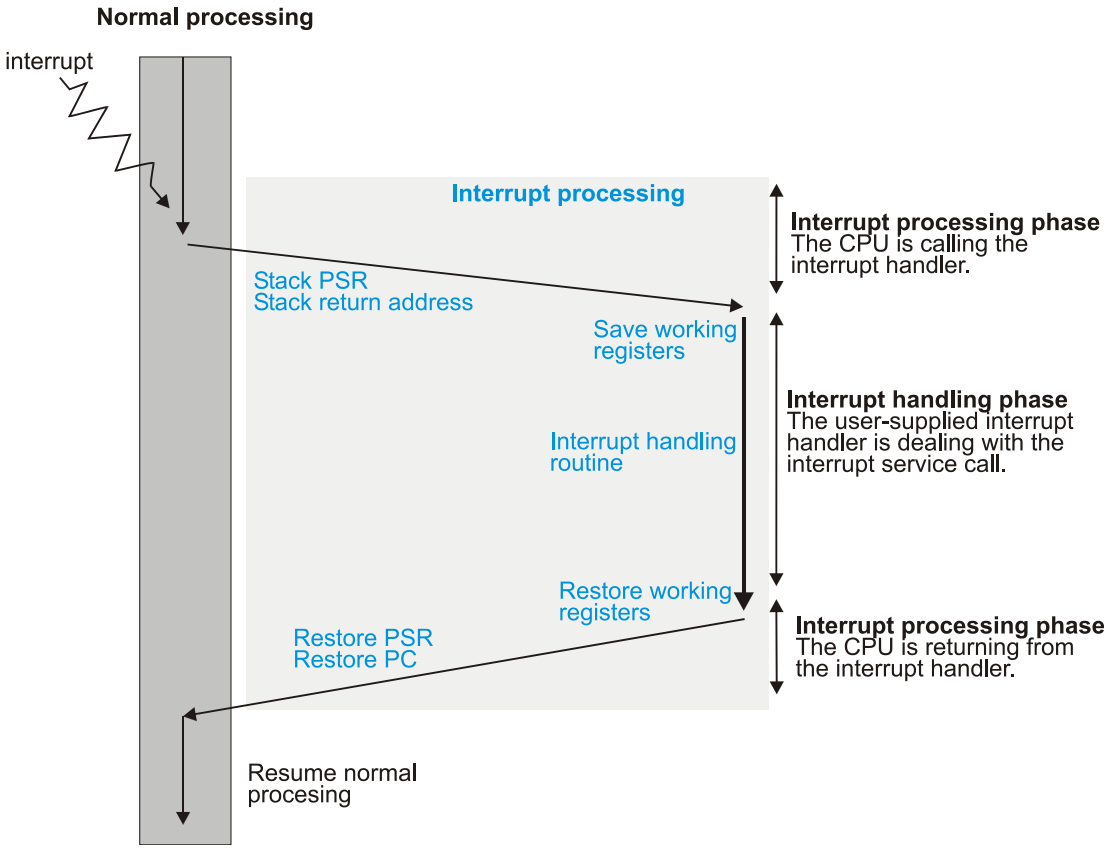


The CPU may or may not respond to an interrupt request. The request may be masked (turned off or disabled) if the CPU is busy executing a higher priority task. It is the function of the operating system to turn on and off responses to interrupts.

When the CPU responds to an interrupt, it must save the current machine context (the program counter, the processor status word and – optionally – the contents of registers). These are often saved on the system stack. The CPU then determines the cause of the interrupt (usually by polling each possible peripheral in turn by reading its interrupt status bit). When the interrupting device is located, control is passed to the appropriate interrupt handler (the operating system usually plays a role in determining the location of the interrupt handler).

After the interrupt has been processed, a return from interrupt is made and the saved registers, status word, and program counter restored. The figure below demonstrates the basic interrupt handling sequence. Note that the processor (via the operating system) must reset the state of the interrupting device to avoid a second (and infinite sequence of) interrupt when the interrupt has been serviced.

Note that interrupts can be nested so that an interrupt can be interrupted.



Examiner's comments

A small number of candidates attempted this question and the evidence shows that a few produced an answer which demonstrated knowledge of the use of the stack mechanism. However, some candidates provided general discussions about interrupts and input-output instead of a structured answer.

Section B

B5

Storage in computer systems is essential. Using examples of their use, briefly explain:

a) Hard disk **(3 marks)**
b) Magnetic tape **(3 marks)**
c) RAM **(3 marks)**
d) USB pen drive **(3 marks)**

Answer Pointers

A brief description of each type of storage was required. Relevant examples of the use of these storage mediums was required. Disk storage is required to store a range of applications and data. Candidates must appreciate development in methods of storage.

Examiner's comments

Answers to this question were good. The evidence shows that candidates were able to appreciate the range of storage mediums. Weaker candidates did not have an understanding of magnetic tape as a storage medium.

B6

Describe the following types of computer network terms:

a) Network address **(4 marks)**
b) MAC address **(4 marks)**
c) IP address **(4 marks)**

Answer Pointers

An explanation of each of the network terms was required. Each of these components contribute to the development and use of computer networks. Candidates were required to show an understanding of contemporary network devices.

Examiner's comments

Candidates focused on the first and third parts of the question. The evidence shows that few answers covered MAC address well. There were detailed explanations of network and IP address.

B7

As a computer technician, you have been asked to provide advice on a range of computing interfaces. Briefly describe and differentiate between:

a) HDMI **(4 marks)**
b) Ethernet **(4 marks)**
c) WiFi **(4 marks)**

Answer Pointers

Each of these devices enable the implementation of a range computing systems. They contribute to the use of computers and networks.

Examiner's comments

Candidates showed a good understanding of the three devices. Good answers included relevant answers of the use of the devices.

B8

Describe the following Operating Systems related terms:

a) Kernel **(4 marks)**
b) Scheduler **(4 marks)**
c) Multiprocessing **(4 marks)**

Answer Pointers

Each function of the OS needed to be clearly described. Some references needed to be made to typical OS currently being used. It was necessary for candidates to understand various processes that take place at the OS level.

Examiner's comments

Good answers were provided by candidates which detailed explanations of these OS terms. The evidence shows that some candidates were not able to demonstrate their understanding of the function of OS in running and managing computer systems.

B9

Explain the purpose of each of the following in distributing data on the internet:

a) World Wide Web **(3 marks)**
b) Router **(3 marks)**
c) HTTP **(3 marks)**
d) Server **(3 marks)**

Answer Pointers

Data distribution is important in a well structured and managed computing environment. Candidates were required to demonstrate a good understanding of the availability of data (and information) in order to carry out relevant tasks.

Examiner's comments

Overall, candidates showed a good understanding of these data distribution techniques. Weaker answers did not fully cover the use of HTTP. The evidence shows that there were some confusions between World Wide Web and HTTP.

B10

Computers need to be protected and made more secure. Describe briefly the following computer security terms:

a) Firewall **(3 marks)**
b) Access Control List **(3 marks)**
c) Pop-up blocker **(3 marks)**
d) Anti-Virus Software **(3 marks)**

Answer Pointers

Secure and reliable Information is becoming increasingly important in organisations. Loss of data can lead to financial difficulties and loss of customers. Candidates must understand the need to take relevant precautions in safeguarding information stored in computer systems.

Examiner's comments

The evidence shows that most candidates were able to provide a good explanation of the various security terms. Good answers included the use and application of the security methods.

B11

a) Differentiate between a program counter and an instruction register. **(4 marks)**
b) Describe the Fetch Execute Cycle during the operation of the CPU. **(8 marks)**

Answer Pointers

Candidates needed to explain the role of each register in executing instructions at processor level. It is important for candidates to have a good understanding of the main registers required in handling instructions and data by the processor of the computer.

Examiner's comments

The evidence shows that answers were often inaccurate and did not demonstrate a good understanding of each register. Candidates would benefit from revising the use of these fundamental registers in computing.

B12

Briefly describe the four layers of the TCP/IP model. **(12 marks)**

Answer Pointers

Brief description of each of the four layers of the TCP/IP model was required.

Examiner's comments

The evidence shows that most candidates were able to identify the four layers of the TCP/IP model. Good answers showed an in depth understanding.