Mark schemes

The following annotation is used in the mark scheme:
; - means a single mark
// - means alternative response
/ - means an alternative word or sub-phrase
**A** - means acceptable creditworthy answer
**R** - means reject answer as not creditworthy
**NE** - means not enough

**Q1.**

Processor management // Allocation of processors // Allocation of processor time // (process) scheduling // thread management;

**A.** Processing management, CPU management

Allocation/management of RAM / memory // allocation of buffers;

Allocation/management of / control of I/O devices/peripherals //

I/O management // device driver management;

File / backing store / secondary store management / access / organisation;

Power / battery management;

Interrupt handling;

**A.** Provision of Application Program Interface / API

**A.** interface between hardware and applications

**A.** Provision / management of (windows in) user interface

**A.** Management of system security

**A.** Answers by example, only one example of each type

**A.** A description of a type of software management but not just "software management". e.g. loading of programs, software installation, registering DLLs.

**A.** Managing network connections but **R.** Network management

**R.** Software management alone unless role of OS in this is clear e.g. installation of new software, updating registry

**MAX 3**

**[3]**

**Q2.**

(a)

|  |  |  |  |
| --- | --- | --- | --- |
| Comp | S/ware | Hardware | Hardwareandsoftware |
| Wireless router |   |   | TICK; |
| Compiler | TICK; |   |   |
| Keyboard |   | TICK; | **A**  TICK; |

**A** crosses used rather than ticks
**R** answers where two columns have been ticked in a single row

**3**

(b)     (i)      Provide an interface between the computer and user;
To manage devices / files / memory;
To provide a virtual machine;
To provide a software platform on which other programs can run // to run application software;
To hide the complexity of the hardware from the user;

**NE** to allow user to use hardware
**R** to execute commands

**Max 1**

(ii)     To allow sharing of run-time code across programs;
To save memory as routines are only loaded when needed;

To provide access to procedures / functions when writing a program;,
To reduce amount of programming required // time taken to write program;

**Max 1**

(c)     (i)      Meets all of the end-user requirements;
Only performs necessary functions;
Can be fitted into existing hardware / software;

**Max 1**

(ii)     More expensive as have to cover production costs;
Not available immediately // have to wait for software to be written;
Less widely tested so more likely to contain bugs;
Lack of 3rd party support;

**NE** (more) expensive
**R** “no testing”

**Max 1**

**[7]**