UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Diploma Advanced Level

MARK SCHEME for the November 2006 question paper

CAMBRIDGE INTERNATIONAL DIPLOMA IN COMPUTING

5218 Further Systems and Software
Maximum mark 90

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began.

All Examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes must be read in conjunction with the question papers and the report on the examination.

The grade thresholds for various grades are published in the report on the examination for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level syllabuses.

- CIE will not enter into discussions or correspondence in connection with these mark schemes.

CIE is publishing the mark schemes for the October/November 2006 question papers for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level syllabuses and some Ordinary Level syllabuses.

UNIVERSITY of CAMBRIDGE
International Examinations

1 (a) (i) -stores the instruction that
-is currently being processed
-splits the binary code into operation code and address
(1 per -, max 2) (2)

(ii) -stores the address (in memory)
-of data to be accessed (from memory)
-instruction/raw data
(1 per -, max 2) (2)

(iii) -stores the address of the next instruction to be accessed
-is incremented (after contents are copied to MAR)
-is altered to allow for jump instructions
(1 per -, max 2) (2)

(iv) -contains a value which is added to the address (in the CIR)
-in order to make the address of the data
-incremented after use so that a set of data can be read one after the other without altering the raw address
(1 per -, max 2) (2)

(b) (i) -a number of processors
-operate together
-so that a set of operations can be carried out simultaneously
(1 per -, max 2) (2)

(ii) -any example that requires large amounts of processing e.g. weather forecasting
-because large quantities of processing are required **in a set time period** (2)

2 (a) (i) **unique** value in the table used to **identify** the record

(ii) key used to access the records in a different order

(iii) an attribute in one table that is a primary key in another table/to provide a link between tables (3)

(b) -reduces duplication of data/no duplication of data
-(improved) data integrity
-allows for different views of the data
-more simple to control access to data
-simpler/faster/easier to access specific data through searches/queries
(1 per -, max 3) (3)

3 (i) -the old system and the new system are run together (until the new system is proven)
-very important application so the costs are worth paying
-time is important in producing results so cannot afford to wait while bugs are corrected
-allows workers to become familiar with new system before changeover
-reduces risk to end product

(ii) -one area of the organization is converted to the new system while the remainder uses the old
-could be one subject/one area of the world
-would mean that effect of any problems would be minimized (and so small that remedial action could be taken)
-allows workers to familiarize themselves with the new system on a rota basis

(iii) -the old system is switched off/the new one takes over immediately
-very risky because the results are so important and
-time dependent
-allows no time for training/finding errors in software solution.
(1 per -, max 3 per section, max 9) (9)

4 (a) -the production of a machine code program/intermediate code which…
-will produce the results intended by the source code
-optimisation reduces the size of the object code by…
-removing any duplicate or redundant instructions…
-which improves speed of execution
(1 per -, max 3) (3)

(b) (i) -linkers join together (compiled) modules of code
-to produce an executable file
-needs to match up address references between modules

(ii) -takes a set of code from storage and copies it into memory
-needs to resolve problems with addresses
-mention of linking loader
(1 per -, max 2 per section, max 4) (4)

5 (a) (i) 01101101 (1 per nibble) (2)

(ii) 0001 0000 1001 (1 for use of 12 bits, 1 for correct answer) (2)

(iii) 6D (1 per digit) (2)

(b) (i) -46 (1 for negative, 1 for 46) (2)

(ii) -(1)00101111/result = +47
-a positive and negative have been added together and the result is positive
-because the larger value was positive.
-there was carry in and out of MSB therefore ignore carry out, (result is correct).
(1 per -, max 1 for either answer, max 2 for discussion, max 3) (3)

6 -workers supervised more closely because of electronic 'spying'
-can be disciplined for lack of work
-can be rewarded for 'hidden' work
-less trade union power
-because workers do not work as a unit
-less pressure on workers in disputes
-call centre type jobs
-clean, well paid
-soul destroying
-need to train to keep abreast of use of technology
-more qualified worker can demand higher rewards
-meetings and video conferencing
-use of emails and other communication
-dangerous tasks made safer
-new job types created
-allows more work to be done in the same amount of time
-use of technology to carry out old tasks in a new way/new data storage and retrieval techniques
(1 per -, max 8) (8)

7 (a) -a computer language used to create multimedia pages
-each page consists of the text to be displayed
-tags providing special instructions about the display
-provides links to files/pages (picture/sound/video/…)
(1 per -, max 2) (2)

(b) -Tags may be used to indicate where
-illustrations are to be inserted into the text
-Tags can be used to change text style
-sizes/fonts
-Tags may be used to change colours of
-backgrounds/text
-Tags may be used to define some text as a link
-or as a hot button/spot
-Links
-provide a fast way of navigating between pages
-Use of different page areas which
-allow different rules in each area/heading and body/makes searching easy
(1 per pair, max 3 pairs, max 6) (6)

8 (a) Input: Any two from touch/radar/proximity/infra red sensors
Output: Any two from alarm/speakers/lights/motors to activate wheels/steering/actuators (4)

(b) (i) -optical sensors
-radar
-used to detect obstacles
Positions determined by
-angular bearing from reference point
-distance from radar
(1 per -, max 2) (2)

(ii) -design must be created using simulation because of large cost of real thing
-testing also simulated because not possible to test in real environment.
(1 per -, max 2) (2)

(c) -factory robot is physically available to people to control it/commands acted upon immediately/need to have immediate action because of proximity to humans
-Mars robot cannot be controlled in real-time because of the time taken for instructions to reach it/instructions need to be sent as a batch and then acted upon and results sent back to operator on earth. (2)

9 (a) -a number of jobs will want to be run at the same time
-processor can only run one job at a time
-in order that the jobs are treated 'fairly'
-the operating system has to have rules to determine the order of execution/make maximum use of resources
(1 per –, max 2) (2)

(b) -order of jobs according to list of priorities
-each job allocated priority according to…
-importance/time already spent on job/need for peripheral devices
-jobs can be in any of three states: ready, running or blocked
-**Ready** Q contains list of jobs waiting for processing in the order in which they should be processed
-HLS handles ready Q and loads jobs
-MLS handles the swapping of data between memory and storage
-LLS moves jobs in and out of running state
-preemptive scheduler has control over what is in running state, non-preemptive simply follows the Q
(1 per -, max 5) (5)

10 (a) -a number of programmers can all work on the same piece of software
-individual expertise can be utilized
-errors are far more easily spotted because…
-each procedure/function is much simpler to solve than the original problem
-individual procedures are far easier to test than a whole project
-library routines can be utilized
-procedure can be used multiple times
-functions are mathematically provable to be correct/faulty
(1 per -, max 4) (4)

(b) **(i)** a variable whose value only applies in a particular procedure

(ii) a variable whose value applies throughout a program

(iii) a value which is applied to a variable within a procedure, and only within that procedure

(iv) the value to be applied is stored in a memory location which is passed to the procedure. Any change will be carried out of the procedure. (4)

(c) -return address placed on stack
-along with values of parameters
-parameters read off stack by procedure
-any returning values placed on stack by procedure
-return to address at top of stack at end of procedure.
(1 per -, max 4) (4)