Version 1.0: 0606

ASSESSMENT and QUALIFICATIONS ALLIANCE

General Certificate of Education

Computing 6511

CPT4 Processing and Programming Techniques

Mark Scheme

2006 examination - June series

Mark schemes are prepared by the Principal Examiner and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation meeting attended by all examiners and is the scheme which was used by them in this examination. The standardisation meeting ensures that the mark scheme covers the candidates' responses to questions and that every examiner understands and applies it in the same correct way. As preparation for the standardisation meeting each examiner analyses a number of candidates' scripts: alternative answers not already covered by the mark scheme are discussed at the meeting and legislated for. If, after this meeting, examiners encounter unusual answers which have not been discussed at the meeting they are required to refer these to the Principal Examiner.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of candidates' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

INSTRUCTIONS TO EXAMINERS

The following forms of notation should be used on candidates' scripts:

- Ticks - To indicate what is accepted as correct or creditworthy, placed in the body of the answer, and on diagrams;
- Underscoring - To identify errors/irrelevance in written answers;
- Crosses - to indicate a wrong answer;
- Brief comments - placed in at suitable points in the body of the text to amplify the marking;
- BOD - means benefit of the doubt and is used where the candidate's answer has been given a mark on the balance of probabilities that the candidate's answer has met the requirements of the mark scheme even though it could be interpreted differently;
- NE - means not enough and is applied to an answer that falls short of what is required;
- O/S - means outside the mark scheme. The candidate's answer is creditworthy but the answer does not match any of the answers on the mark scheme for the particular question. Nevertheless a mark is awarded;
- C/F - means carried forward. This arises when a candidate offers an answer that is not creditworthy in that part of the question but is creditworthy in a later part of the same question. The mark is carried forward to the part of the question that is creditworthy;
- C/B - means carried back. This is similar to a carry forward but the mark is carried back to an earlier part of the question.
- T/O - means talked out. The candidate's answer is contradictory.
- F/T - means followed through. If the candidate made a mistake in the earlier part of an answer, mark the answer using the correct method on their answer from the earlier part.

The following notation is used in the mark scheme

- ; - means a single mark;
- **A** - means an acceptable creditworthy answer;
- **R** - means reject answer as not creditworthy.
- **I** - ignore
- / - means alternative word or sub-phrase
- / / - means alternative answer

General Rules for marking

Ignore Abbreviations
Ignore Brand Names

1 (a) 00011011;

1;1101101; **3 marks**

(b) 10000;1000; Allow FT from (a)
Stop marking when an error occurs **2 marks**

(c) (i) Carry Bit; **R** Overflow bit **1 mark**

(ii) For multi-word/byte operations//test for overflow **1 mark**

(d) 1B;
ED;
Allow FT **2 marks**

Total 9 marks

2 (a) Server provides a resource/the Internet/a database/file/application/ CD ROM/printer;
Within a network;
Client computer requests the service;
and waits for the response;
I any reference to the user Max 3 **3 marks**

(b) (i) More than one process; **A** program/application/task/job (being executed) concurrently/apparently simultaneously/ apparently at the same time; **2 marks**

(ii) Use of a single copy of a program;
To process several sets of data;

A Threads share data;
A Separate paths of execution; through a single program; Max 2 **2 marks**

(c) (i) Software request causes an event; **A** external change causes an event
The event causes a program/ procedure/ function to execute;
I References to user **2 marks**

(ii) Mouse Click// mouse movement// key pressed// record read/ written// any external/internal device requires attention; **1 mark**

Total 10 marks

3 (a) (i) 01000001; **A** 41//65 **1 mark**

(ii) 01000101; **A** 45//69 **1 mark**

(iii) 00001111; **A** 0F//F//15 **1 mark**

(b) (i) Contents/offset; of Index Register; added;
to a (base) address; **A** value / number / constant / operand **4 marks**

(ii) Where a contiguous block of memory is to be referenced//Array manipulation;

A vectored mechanism
R relocation of code **1 mark**

Total 8 marks

4 (a) Part of the hard disk is allocated to be used as virtual memory;
Contents are copied into main memory as required;
Partially loaded programs may be executed;
Virtual address space can exceed physical address space; Max 3 **3 marks**

(b) Main Memory is divided into fixed sized pages/frames;
Program subdivided into same sized pages;
Pages are swapped/loaded as needed;
Page table keeps track of pages;

A Paging is a method of implementing virtual memory;

Do not give same point in (a) and (b) Max 2 **2 marks**

(c) The memory that is unallocated/ available/free;
Is used by the operating system to allocate memory to processes/running programs;
Is used when a process/running program requires memory dynamically;
Max 2 **2 marks**

Total 7 marks

5 (a) (i) 271; **1 mark**

(ii) The required item might be the 271st one/last one/ not be present//
Every item accessed; **1 mark**

(b) (i) 9; **1 mark**

(ii) Each comparison halves the number of items to be accessed//
271 lies between 2^8 and 2^9; **1 mark**

(c)

Count1	Count2	Temp	A				
			[1]	[2]	[3]	[4]	[5]
-	-	-	23	45	16	12	31
1	1						
	2	45		16	45		
	3	45			12	45	
	4	45				31	45
2	1	23	16	23			
	2	23		12	23		
	3						
	4						
3	1	16	12	16			
	2						
	3						
	4						
4	1						
	2						
	3						
	4						

1 mark

1 mark

1 mark for Count1
1 mark for Count2
1 mark for Temp **5 marks**

(ii) (bubble) sort the items into ascending order; **1 mark**

(ii) Reduce the number of tests each pass// stop when no swaps occur during a pass//
Add a flag NoSwaps to indicate when no swaps occur// change loop control to Repeat until no swaps// sort variable sized array; **1 mark**

Total 11 marks

6 The assembly language instruction MOV or MOVE may be used in place of LOAD, LD, LDA, STORE, ST, STA.
Immediate addressing must be used for constants but could be indicated by annotation e.g. load register with 0. Indicate any omissions
A pseudocode statement gets no mark. Order of statements is important.
Ensure that loop continues while count <= 5 or count < 6.
If subtract is used in place of CMP check value of count is not destroyed.
Accept relative addressing for jumps.
No marks for jumps not altering flow of control.

initialise X;
initialise Count;
start of loop;
compare;
correct loop back;
add Count to X;
increment Count;
correct exit;

penalise only once for incorrect columns/syntax

Total 8 marks

<u>Example 1</u>

```
        LD register,#0
        ST register, X          initialise X                    1 mark
        LD register, #1
        ST register, Count      initalise count                 1 mark
label1:                         start of loop                   1 mark
        LD register, Count
        CMP register, #5        compare                         1 mark
        BEQ label2              correct branch to label
        ADD register,X
        ST register,X           add count to X                  1 mark
        LD register, Count
        INC register or ADD register,#1
        ST register,Count        increment Count                1 mark
        JMP label1               correct loop back              1 mark
label2:                          correct exit                   1 mark
```

<u>Example 2</u>

```
X:      EQUW    0       initialise X                            1 mark
Count:  EQUB    1       initalise count                         1 mark
label1:                 start of loop                           1 mark
        LD      register, Count
        ADD     register,X
        ST      register,X      add count to X                  1 mark
        INC     Count           increment Count                 1 mark
        LD      register,Count
        CMP     register, #6    compare                         1 mark
        BNE     label1          correct loop back               1 mark
                                correct exit                    1 mark
```

7 (a) Halve the time to perform an operation;
A Operations performed more quickly; **1 mark**

(b) Increase the number of bits transferred at any one time from 16 to 32//
Double the number of bits transferred at any one time; **1 mark**

(c) Increase the number of memory addresses; from 2^{24} to 2^{32}; **2 marks**

Total 4 marks

8 (a) male (richard);
parent (john, richard);
parent (kylie, richard);
Penalise once for wrong case **3 marks**

(b) michael; linda;
Penalise once for wrong case **2 marks**

(c) IF male (Dad) AND parent (Dad, Child)

1 mark for IF (or :-)
1 mark for AND (or ,)
1 mark for both male (Dad) & parent (Dad, Child) **3 marks**

Total 8 marks