THE AUSTRALIAN NATIONAL UNIVERSITY

Mid Semester Examination, April 2010

COMP2300 / COMP6300
(Introduction to Computer Systems)

Writing Period: 1 hour duration

Study Period: 0 minutes duration

Permitted Materials: One A4 page with notes on both sides.
NO calculator permitted.

Questions are NOT equally weighted.
This exam will contribute 20% (redeemable) to your final assessment.

The questions are followed by labelled, framed blank panels into which your answers are to be written. Additional answer panels are provided (at the end of the paper) should you wish to use more space for an answer than is provided in the associated labelled panels. If you use an additional panel, be sure to indicate clearly the question and part to which it refers to.

The marking scheme will put a high value on clarity so, as a general guide, it is better to give fewer answers in a clear manner than to outline a greater number in a sketchy, half-answered fashion. The Appendix contains a table with powers of 2 values in decimal.

Please write clearly – if we cannot read your writing you may lose marks!

Student Number:

Official use only:

Q1 (13)	Q2 (17)	Total (30)

QUESTION 1 [13 marks]

(a) Assume memory addresses `0x02000411` to `0x02000415` contain the following 8-bit binary values:

Address	0x02000411	0x02000412	0x02000413	0x02000414	0x02000415
Binary value	0010 1011	1001 0011	1100 1001	0010 0111	0001 0101

Assume the C declaration of `short int x, y;` (with `sizeof(x)` and `sizeof(y)` both being 2), `&x = 0x02000412` and `&y = 0x02000414`, and the data storage is *big endian*.

(i) What would be printed by the following C statement?

```
printf("Values for x+y: %x %o\n", x+y, x+y);
```

Clearly show how you derive your answers.

QUESTION 1(a)[i]	**[2 marks]**

(ii) What would be printed by the following C statement?

```
printf("Value for (x+y)>>8: %d\n", (x+y)>>8);
```

where `>>` is the bitwise right-shift operator (which presenrves the sign of the 2's complement value on the left). Clearly show how you derive your answer.

QUESTION 1(a)[ii]	**[1 mark]**

Question 1 (continued)

(b) The IEEE single-precision floating-point standard is: 1 bit sign, 8 bits exponent with a bias of 127, and the remaining 23 bits are the significand (mantissa), with an implicit leading bit. What floating point number is represented by the following 32-bit number (given in hexadecimal representation):

`0xC2940000`

For full marks, your answer should be expressed as a single decimal number accurate to four decimal digits.

QUESTION 1(b)	**[2 marks]**

(c) Negative integers can be represented by reserving a bit to represent the sign (for example, for 8-bit: 10000000, 10000001, 111111111 could represent -0, -1 and -127). State two advantages of two's complement arithmetic over this representation.

QUESTION 1(c)	**[1 mark]**

Question 1 (continued)

(d) In the IEEE single-precision floating-point standard, the three components of the number have the following ordering:

31	30							23	22																						0
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m
sign	exponent								significand																						

What advantage would this ordering have over other possible orderings?

QUESTION 1(d) **[1 mark]**

(e) A half-adder circuit is given by the following table:

Inputs		Output	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Outline how it could be constructed using basic gates (AND, OR, XOR, NOT, etc). *Note: it is not necessary to give a formal circuit diagram in your answer.*

QUESTION 1(e) **[2 marks]**

Question 1 (continued)

(f) Briefly describe the function of the *bus* in a computer.

QUESTION 1(f)	**[1 mark]**

(g) Briefly describe the function of the *control unit* within the Central Processing Unit.

QUESTION 1(g)	**[2 marks]**

(h) Variable instruction length is a characteristic of CISC computers. In contrast, the original RISC computers had a fixed instruction length. What was the primary advantage of having a fixed instruction length?

QUESTION 1(h)	**[1 mark]**

QUESTION 2 [17 marks]

(a) Write C statements to declare an single-precision floating point variable `x` initialized to `1.0`, and a floating point pointer `px` initialized to the address of `x`.

QUESTION 2(a)	**[1 mark]**

(b) Consider the following C program `foo.c`

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
   int a[64], i;
   int n = atoi(argv[1]);
   a[0] = 0;
   for (i = 1; i < n; i++) {
     a[i] = a[i-1] + 2 * i;
   }
   printf("%4d\n", a[n-1]);
   return 0;
}
```

Suppose the program was compiled and linked into an executable program called `foo`.

(i) Write the output produced by the command `./foo 5`, clearly indicating spaces (with a '␣'). *Hint:* in this case, the variable `n` will be assigned the value of 5.

QUESTION 2(b)[i]	**[2 marks]**

(ii) Suppose n represents an integer and $0 \leq n < 64$. In terms of n, state what the command `./foo` n produces.

QUESTION 2(b)[ii]	**[1 mark]**

Question 2 (continued)

(iii) What is the main limitation of the above program, and briefly describe the best way to rectify it.

QUESTION 2(b)[iii] **[2 marks]**

(iv) Even after this limitation is removed, the above program can still abort with a 'segmentation violation' error. Briefly describe how this can occur, and indicate how you would rectify this.

QUESTION 2(b)[iv] **[2 marks]**

(c) Write a sequence of C statements that (i) defines a structure called `safestr` consisting of the fields `s` (of type pointer to `char`) and `len` of type integer, (ii) declares a variable `ss` of type `struct safestr`, and (iii) initializes its `s` field to 64 bytes of dynamically allocated memory and its `len` field to 64.

QUESTION 2(c) **[2 marks]**

Question 2 (continued)

(d) Consider the following function definition.

```
void hist(int n, int p[], int k, int h[]);
// post: h[i] is the number of occurrences of i
//       in p[0], p[1], ...., p[n-1]
//       for each i such that 0 <= i < k
// example: if n=6, p = {0,1,3,1,1,3} and k=4, h = {1,3,0,2}
```

Suppose that the C file `hist.c` contained am implementation of the `hist()` function, and the C program `histogram.c` calls this function.

(i) Write an implementation of the `hist()` function.

QUESTION 2(d)[i] **[3 marks]**

(ii) C provides an `assert()` facility. Illustrate how the facility could be used in the above function.

QUESTION 2(d)[ii] **[1 mark]**

Question 2 (continued)

(iii) Write the contents of a header file, `hist.h`, which, if included at the top of `hist.c` and `histogram.c`, would ensure that the call of `hist()` in `histogram.c` was consistent with its implementation in `hist.c`.

QUESTION 2(d)[iii]	**[1 mark]**

(iv) Suppose the system was compiled using `gcc -c hist.c` and `gcc -c histogram.c` and linked into an executable program using `gcc -o histogram histogram.o hist.o`. Briefly explain how the reference of the external symbol `hist` is resolved into an address during the linking stage.

QUESTION 2(d)[iv]	**[2 marks]**

Additional answers to QUESTION ___(___)[___]

Additional answers to QUESTION ___(___)[___]

Additional answers to QUESTION ___(___)[___]

Additional answers to QUESTION ___(___)[___]

Additional answers to QUESTION __(__)[__]

Appendix

x	2^x
-5	0.03125
-4	0.0625
-3	0.125
-2	0.25
-1	0.5
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536

Table 1: Powers of 2 in decimal