0931

Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

Summer Examinations, 2005/2006

Exam Code(s)	1BS1, 1CS1, 1EH1, 1EL1, 1ER1.
Exam(s)	**First Science**
Module Code(s)	CS102
Module(s)	**Computer Science**
Paper No.	1
Repeat Paper	
Special Paper	
External Examiner(s)	Prof. D. Bell;

Internal Examiner(s)	Prof. G. Lyons;
	Dr Ray Butler;
	Dr. Götz Pfeiffer;
	Mr. Aidan Power;
	Dr. B. Gleeson.

Instructions:	Attempt *SIX* questions, at least *SIX* from each section. Use a SEPARATE ANSWER BOOK for each section. All questions carry the same marks.
Duration	*THREE* hours
No. of Answer books	3
Requirements:	____________
Handout	____________
MCQ	____________
Statistical Tables	____________
Graph Paper	____________
Log Graph Paper	____________
Other Material	____________
No. of Pages	5
Department(s)	INFORMATION TECHNOLOGY, MATHEMATICS, EXPERIMENTAL PHYSICS, MATHEMATICAL PHYSICS

SECTION A

1. *Answer all parts of this question:*

   (a) Using 2's-complement representation, evaluate the following:

   i) $37_{10} + (-24_{10}) = ?$

   ii) $187_{10} + (-107_{10}) = ?$

   (b) Using the usual ANSI/IEEE 754−1985 standard for single-precision real (floating point) numbers, what would be the representation of the decimal value 93.175781_{10}?

   (c) Explain how the Hamming Code works. Determine the Hamming Code for the 7−bit binary equivalent of 53_{10}.

2. *Answer all parts of this question:*

   (a) State both of de Morgan's Laws for Boolean operators, and prove either law using a truth table.

   (b) By means of a truth table and a circuit diagram, design a *Half Adder* circuit, which adds two input bits and outputs two bits called SUM and CARRY.

   (c) Construct the truth table for a *Full Adder*, and show how its circuit diagram can be constructed from two *Half Adder* blocks.

3. *Answer all parts of this question:*

   (a) Discuss six of the different components that you would expect to find within a simple Central Processing Unit (CPU).

   (b) Describe the operation of a two-pass assembler.

   (c) Write an assembly language programme for the CS1 CPU, which takes in a list of 10 numbers (from the file data.in), and writes them out (to the file data.out) in descending order.

   The list of CS1 assembly language instructions is given below:

01	READ	06	SUBTRACT	11	JPA	16	SBX
02	WRITE	07	MULTIPLY	12	JNA	17	SJX
03	LOAD	08	DIVIDE	13	LDX	18	JZX
04	STORE	09	JUMP	14	STX	19	DJZ
05	ADD	10	JZA	15	ADX	20	HALT

SECTION B

4. (a) Write a statement or comment, as appropriate, to accomplish each of the following:

   i) State that a program will calculate the product of three integers.

   ii) Define the variables `x`, `y`, `z` and `product` to be of type `int`.

   iii) Prompt the user to enter three integers, please.

   iv) Read three integers from the keyboard and store them in variables `x`, `y` and `z`.

   v) Compute the product of `x`, `y` and `z` and assign the result to the variable `product`.

   vi) Print "`The product is` " followed by the value of the integer variable `product`.

   (b) Describe a control structure that allows parts of a programme to be executed conditionally. What effect does it have on the flow of control?

   (c) Write a function which, given an integer `n`, returns `-1`, if `n` is negative, or `1`, if `n` is positive or `0` otherwise.

5. (a) Explain the following terms:

   i) *operator*, ii) *comment*, iii) *variable*, iv) *expression*, v) *statement*.

   (b) Explain the concepts of precedence and associativity in a programme. Describe the precedence and associativity of `+`, `-`, `*`, `%`, and `=`.

   (c) Indicate the order of evaluation of the operators in each of the following statements, and determine the resulting value of `x`:

   i) $\mathtt{x} = 1 + 3 * 6/2 - 7$; and ii) $\mathtt{x} = 2\%2 + 2 * 2 - 2/2$;

6. (a) Write a programme that prompts the user to enter an integer, stores this value in the variable `n`, and then prints `n` rows of `3*n` stars. If `n` is 3 the output should be:

```
*********
*********
*********
```

   (b) Write a programme that, depending on the definition of `N`, prints a triangular shape of `N` lines of stars. If `N` is 3 the output should be:

```
***
 **
  *
```

   Describe the important parts of your programme, and explain briefly why the programme does the right thing.

SECTION C

7. Describe briefly, using diagrams and/or examples where appropriate, ANY THREE of the following:

   (a) Magnetic and Optical Storage Devices;

   (b) Operating Systems;

   (c) Components of the CPU;

   (d) Time-sharing and deadlock;

   (e) Local Area Networks;

8. (a) Explain briefly how user-defined structured variables can be useful giving two examples.

   (b) Write a commented programme, using the following structure type:

```
# include <stdio.h>
typedef struct
{
      float re, im;
} complex;
main(void)
{
      complex a, b, product;
```

   to input two complex numbers, and output their product. Use the functions **void** input(**complex** *x) and **void** multiply(**complex** *x, **complex** *y, **complex** *z).

9. (a) Explain briefly what a pointer is, and why it is useful.

   (b) The following code is part of a C programme:

```
int a, b, c, *x, *y, *z;
z = &b;
a = 20;
b = a + 9;
x = &c;
*x = a/4 + b;
y = z;
*y += b;
b -= c;
```

   After the code has been entered, what values will a, b, c, *x, *y and *z have? Explain how you arrive at your answers.

(c) The following programme inputs ten numbers and stores them in an array, and then bubble sorts the elements into ascending order before outputting them again:

```
# include < stdio.h >
main(void)
{
    int i, array[10];
    void bubblesort(int *)
    float min(int, int);
    printf(“Please input ten integers.\n”);
    for (i = 0; i < 10; i++)
        scanf(“%d”, &array[i]);
    bubblesort(array);
    printf(“The sorted array is as follows:\n”);
    for (i = 0; i < 10; i++)
        printf(“%d\n”, array[i]);
}
```

Write definitions for **int** min(**int**, **int**) and **void** bubblesort(**int** *).