King's College London

UNIVERSITY OF LONDON

This paper is part of an examination of the College counting towards the award of a degree. Examinations are governed by the College Regulations under the authority of the Academic Board.

B.Sc. EXAMINATION

CP1710 Computing for Physical Sciences

Summer 2004

Time allowed: THREE Hours

Candidates should answer no more than SIX parts of SECTION A, and no more than TWO questions from SECTION B.
No credit will be given for answering further questions.

The approximate mark for each part of a question is indicated in square brackets.

You must not use your own calculator for this paper.
Where necessary, a College calculator will have been supplied.

TURN OVER WHEN INSTRUCTED

Physical Constants

Permittivity of free space	ϵ_0	=	8.854×10^{-12}	$\mathrm{F\,m^{-1}}$
Permeability of free space	μ_0	=	$4\pi \times 10^{-7}$	$\mathrm{H\,m^{-1}}$
Speed of light in free space	c	=	2.998×10^{8}	$\mathrm{m\,s^{-1}}$
Gravitational constant	G	=	6.673×10^{-11}	$\mathrm{N\,m^2\,kg^{-2}}$
Elementary charge	e	=	1.602×10^{-19}	C
Electron rest mass	m_e	=	9.109×10^{-31}	kg
Unified atomic mass unit	m_u	=	1.661×10^{-27}	kg
Proton rest mass	m_p	=	1.673×10^{-27}	kg
Neutron rest mass	m_n	=	1.675×10^{-27}	kg
Planck constant	h	=	6.626×10^{-34}	J s
Boltzmann constant	k_B	=	1.381×10^{-23}	$\mathrm{J\,K^{-1}}$
Stefan-Boltzmann constant	σ	=	5.670×10^{-8}	$\mathrm{W\,m^{-2}\,K^{-4}}$
Gas constant	R	=	8.314	$\mathrm{J\,mol^{-1}\,K^{-1}}$
Avogadro constant	N_A	=	6.022×10^{23}	$\mathrm{mol^{-1}}$
Molar volume of ideal gas at STP		=	2.241×10^{-2}	$\mathrm{m^3}$
One standard atmosphere	P_0	=	1.013×10^{5}	$\mathrm{N\,m^{-2}}$

Where necessary, you may assume that the following C functions are declared in the library `math.h`

```
double cos( double x);  // returns the cosine of x (x in radians)
double exp( double x);  // returns the exponential of x
double sqrt( double x); // returns the positive square root of x
                        // and an error occurs if x < 0
double log( double x);  // returns the natural logarithm of x
                        // and an error occurs if x <= 0
```

Ensure that any C code that you write is clearly laid out, and contains sufficient comments for the reader to understand the code.

SECTION A – Answer SIX parts of this section

1.1) State the rules that must be satisfied by a valid name for a *C* variable. Hence explain which of the following names are not valid for *C* variables.

```
CHAR
one-fifth
phone#
ready2run
return
_true
2big
```

[7 marks]

1.2) Inside a *C* function the variables `a, b` and `c` are declared with the following statements:

```
long a;
double b;
long double c;
```

State the scope of these variables, and specify the type of data that can be stored by each variable.

[7 marks]

1.3) Give examples of two different *preprocessor directives*, and explain briefly their purpose.

[7 marks]

1.4) Describe the operation of a simple `while` loop and explain how it differs in practice from a `do...while` loop. Show how a `while` loop could be re-written as a particular form of `for` loop.

[7 marks]

1.5) A structure of type `atom` is declared and initialised using the following *C* statements.

```
struct atom { int p;          // Number of protons
              int n;          // Number of neutrons
              char name[20];  // Element name
              char symbol[3]; // Chemical symbol
              float mass;     // Mass in a.m.u.
            };
```

Write a few lines of *C* code that will:

(i) declare a variable `cobalt` that is a structure of type atom, initialised so that the members of the structure store the values `27, 32, "Cobalt", "Co"` and `58.933` respectively.

(ii) print out following information, including appropriate values derived from the structure, in the format shown here.

```
Element:
Symbol:
Atomic number:
Mass number:
Mass in a.m.u:
```

[7 marks]

Note: A full `main()` program is **not** required here.

1.6) The *C* function `Unit_vector()` is intended to return to a calling routine the components (u, v) of a unit vector that is parallel to a two-dimensional vector (x, y). It follows that

$$u = \frac{x}{\sqrt{x^2 + y^2}} \qquad \text{and} \qquad v = \frac{y}{\sqrt{x^2 + y^2}}$$

The function is declared as follows:

```
void Unit_vector( float x, float y, float *u, float *v);
```

Provide a suitable definition for the function `Unit_vector()`.

Assume that the `float` variables `x`, `y`, `u` and `v` have all been declared in a `main()` *C* program. Write down the single *C* statement that can then be used to load the variables `u` and `v` with the components of the unit vector parallel to vector (x, y).

[7 marks]

1.7) Write down the output that is generated by the following *C* program, reproducing carefully the layout that the program will produce.

```
#include <stdio.h>  // needed for I/O
int main()
{  int invalue, max=4, diff=1, outvalue=0;
   for (invalue=1; invalue<=max; invalue++, diff += 2)
   {
      outvalue += diff;
      printf("\n invalue=%2d, outvalue=%3d", invalue, outvalue);
   }
   return 0;
}
```

[7 marks]

1.8) A *C* program initialises the variables `a, b, c` and `x` in the following way.

```
short a=10, b=7, x;
short *c=&b;
```

Determine the values of `x` after each of the *C* statements below has been executed.

```
x = (a%b) + (a/b);
x = (a<b)? a-b: *c+a;
x = (++a) + (*c);
```

[7 marks]

SECTION B – Answer TWO questions

2) The binomial coefficient $\binom{n}{r}$ gives the number of different ways in which r items may be chosen from a set of n items. The binomial coefficient is given by the formula

$$\binom{n}{r} = \frac{n!}{(n-r)!\,r!} = \frac{n(n-1)(n-2)\ldots(n-r+1)}{r(r-1)(r-2)\ldots\times 2\times 1} \tag{1}$$

From this definition, it is clear that the binomial coefficient also has the following properties.

$$\binom{n}{r} = \binom{n}{n-r} \qquad \text{and} \qquad \binom{n}{n} = \binom{n}{0} = 1 \tag{2}$$

since, by definition, $0! = 1$.

a) Define a C function that will accept non-negative *integer* values for n and r as arguments, and will return a value of type `double` for the binomial coefficient $\binom{n}{r}$ given by equation (1) above. The function should also issue a warning message, and return the value 0.0, when $r > n$.

[15 marks]

Hint: You may find it helpful to use the expression on the far right side of equation (1), instead of calculating the factorials that appear in the middle expression. The two properties of the binomial coefficient given by equation (2) may also help to make your function run more efficiently.

b) Write a `main()` C program that will carry out the tasks listed below.

(i) Prompt the user to enter a value for n, the number of items in the set, rejecting invalid values and prompting again if necessary.

(ii) Display on the screen the value of the binomial coefficents for all non-negative values of $r \leq n$, using one line of output for each value of the binomial coefficient. The first output line should be in the format

```
Binomial( n,  0) = 1.0
```

and all subsequent lines should follow a similar pattern.

[15 marks]

3a) Describe the **bisection** method for locating a real root of the equation $f(x) = 0$ for x such that $x_0 < x < x_1$, and where $f(x)$ is a continuous real function, with $f(x_0) < 0$ and $f(x_1) > 0$.

[6 marks]

b) Explain briefly what is meant by the term *linear convergence* in the context of a root-finding algorithm, and show that the bisection method has this property. Estimate the accuracy with which a root of $f(x) = 0$ can be located after 20 iterations, assuming that $x_1 - x_0 = 3$.

[8 marks]

c) Explain how the bisection method could be used to find both roots of the equation

$$x^2 - \cos x = 0$$

and suggest a suitable starting interval.

[4 marks]

d) Write a C program to find, and print out, both roots of the equation to an accuracy of 3 decimal places.

[12 marks]

4a) Define a function in C that will return the value of the expression

$$f(x) = \frac{1 - \exp\left(-x^2\right)}{x^2}$$

for any real value of x. Comment in detail on how your function deals with any special case that might arise.

[10 marks]

b) Describe the **trapezoidal rule** for estimating the area under a curve. What are the main tasks that a C program must carry out when implementing the trapezoidal rule for this purpose?

[10 marks]

c) Write a C program that will use the trapezoidal rule to calculate numerically, and then print out, the value of the integral

$$\int_0^3 f(x)\,dx$$

for the function $f(x)$ defined above.

[10 marks]

5) In an experiment to measure radioactive decay, the data are stored as formatted values in a plain text file named `decay.txt`. The first three lines of the file are as shown below:

```
1.20
94620
94431
```

The first line of the file contains the time interval between measurements, and is a real value measured in seconds. Measured counts of radioactive decays for 1000 successive time intervals are then stored, as one integer value per line.

Write a *C* program to accomplish the following tasks:

(i) Read the time interval and measured counts from the file `decay.txt`, storing them in appropriate variables.

(ii) Calculate the average value of the count rate, and the time at which the count rate first falls to half its initial value. Display both these pieces of information on the screen, with an appropriate message.

(iii) Calculate an error estimate for each measured count reading, assuming that the error on a measured value of n counts is $\sqrt{n}$.

(iv) Calculate the natural logarithm of every valid value of both the measured count and its associated error. For invalid values, the logarithm should be set to zero.

(v) Write out a new file of processed data, called `decay_log.txt`, which should contain the measurement time interval on the first line, and then pairs of logarithmic values for count and error on subsequent lines. Ensure that these two columns of output data are separated by at least one space.

[30 marks]