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 2005

Time allowed: THREE Hours

Candidates should answer all 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

2005 ©King's College London

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 floor( double x); // returns the nearest integer below x
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 all SIX parts of this section

1.1) Identify which seven of the following names are not valid for *C* variables, giving a reason in each case.

`alpha`	`ID#`
`BREAK`	`string`
`continue`	`touch&go`
`Euro`	`two`
`fifty%`	`wait4keypress`
`GB£`	`_line`
`half-way`	`5perweek`

[7 marks]

1.2) 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
#define NMAX 10
int main()
{
   unsigned int n, m=5;
   int value[NMAX]={0};
   char label[]="data";
//
   for (n=0; n<NMAX; n++)
   {
      value[n]= n % m;
      printf( "%s[%2u]=%2d\n", label, n, value[n]);
   }
   return 0;
}
```

[7 marks]

1.3) A *C* program initialises the variables `u, v, w` and `x` in the following way.

```
float u=3.0, v=4.0, x=5.0;
float *w=&u;
```

Determine the values of `x` after the execution each of the *C* statements below.

```
x += v*(*w);
x = (u<v)? u/v: u*v;
x = (--u)*(++v);
```

[7 marks]

1.4) What information do the following *C* statements provide to the compiler?

```
void Print_message( const char *message, int colour);
struct book{ char author[30];
             char title[40];
             unsigned short pages;
             unsigned short year;
           };
struct book mynovel;
```

[7 marks]

1.5) Provide the *C* code needed to define a function `Swap()`, which should have pointer arguments, that can be used to interchange the values held in two variables of type `float`.

The variables `x` and `y` are declared and initialised in a `main()` *C* program as follows:

```
float x=5.0, y=7.0;
```

Write down the single *C* statement that you would use to interchange the values held in variables `x` and `y`.

[7 marks]

1.6) Write a short *C* program that will carry out the following tasks:

(i) repeatedly prompt the user to enter a real number, exiting when the user enters the value 0.0, and

(ii) display on screen the running total of all numbers entered so far.

[7 marks]

SECTION B – Answer TWO questions

2) Readings from a detector are collected at one-second time intervals over a period of exactly 24 hours, and are stored in a plain text file named `alldata.txt` as a set of real values, with one value per line. The readings lie in the range from 0 to 10.0 V, but the vast majority of them correspond to low-level background noise. The signals of interest are pulses of a few seconds duration that are invariably above 4.0 V.

a) Write a `main()` *C* program to carry out the following tasks:

(i) Read the data from the specified file into a one-dimensional array named `rawdata`. Ensure that the array is large enough to hold all the data in the file.

(ii) Search through the `rawdata` array to find all cases where the signal value is greater than or equal to 4.0 V. For every instance where the signal satisfies this condition, write out to a file, named `signaldata.txt`, a single line of information, in the style shown below, that specifies the time (in seconds) from the start of the recorded data, and the value of the signal at that time. For example:

```
Time=1240 s, Signal=4.3 V
```

(iii) For all cases where the signal exceeds the threshold value, count the number of signal readings in each one-volt range from 4.0 V up 10.0 V, so that a histogram of the signal height distribution can be obtained. Write the information about each one-volt range to the display, using the example format shown below:

```
223 readings of signal S in range 4.0 <= S < 5.0 V
```

[25 marks]

b) Comment briefly on any important choices you had to make in designing the program.

[5 marks]

3a) The variables x and y are real numbers related by the equation

$$x = \left(\frac{y-1}{y+1}\right) \tag{3.1}$$

Define a function in C that will accept a value of y as an argument and return the corresponding value of x to the calling routine. Ensure that, when $y = -1$, the function returns the value 0.0.

[5 marks]

b) By rearranging equation (3.1) above, it is easy to see that

$$y = \left(\frac{1+x}{1-x}\right)$$

and, for x in the range $-1 < x < 1$, it can be shown that

$$\ln(y) = \ln\left(\frac{1+x}{1-x}\right) = 2x + \frac{2x^3}{3} + \frac{2x^5}{5} + \frac{2x^7}{7} + \ldots + \frac{2x^{2n+1}}{2n+1} + \ldots \tag{3.2}$$

so this series provides a way of evaluating natural logarithms.

Define a function in C that will return the value of $\ln\left(\frac{1+x}{1-x}\right)$ using the series expression given above. The function you define should have the following declaration:

```
double Ln_ratio( double x, int n);
```

where the parameter `n` specifies the number of terms to be used in the series approximation.

[12 marks]

c) Write a `main()` C program to carry out the following tasks:

(i) Repeatedly prompt the user to supply a positive value for the variable `y`, exiting when zero or a negative value is supplied.

(ii) Calculate the corresponding value of x by calling the function you wrote in answer to part (a).

(iii) Print out the value of $\ln(y) = \ln\left(\frac{1+x}{1-x}\right)$, as calculated by the C function `Ln_ratio()`, assuming 10 terms will be included in the series approximation.

(iv) Calculate $\ln(y)$ using the C library function `log()`, and print this value on the line following the result of the series approximation.

[8 marks]

d) Estimate how many terms would be required to calculate $\ln(2)$ accurately to 4 decimal places using the series approximation given in equation (3.2).

[5 marks]

4a) Describe the **trapezoidal rule** for estimating the area under a curve.

[6 marks]

b) The following definition of a *C* function provides one possible way of using the trapezoidal rule to estimate the area under a curve described by the *C* function `f()`. (The line numbers at the left of each line are to aid identification, and are not part of the *C* code.)

```
 1  float f( float x);
 2  float Trapezoidal( float low, float high, int num_strips)
 3  {
 4     float width= (high-low)/(float) num_strips;
 5     float area=0.0;
 6     float x;
 7     for ( x=low; x<high; x += width)
 8     {
 9       area += 0.5 * width * (f(x) + f(x+width));
10     }
11     return area;
12  }
```

For each *C* statement in the code given above, supply a detailed comment that accurately describes the purpose of the statement. Explain briefly how the function definition relates to your description of the trapezoidal rule.

[14 marks]

c) The function `Trapezoidal()` as defined above is not very efficient computationally. Identify two simple modifications that can be made to the code, and explain carefully how these modfications will make the calculation significantly more efficient.

Rewrite the function `Trapezoidal()` to incorporate the two changes you recommend.

[10 marks]