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

CP/1710 Computing for Physical Sciences

Summer 2002

Time allowed: THREE Hours

Candidates must answer SIX parts of SECTION A, and TWO questions from SECTION B.

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

Fundamental constants

pi $\pi = 3.1415926$

Where necessary, you may assume that the following *C* functions are defined in the library <math.h>

```
double sqrt( double x); // returns the square root of x
double exp( double x);  // returns the exponential of x
double cos( double x);  // returns the cosine of x (x in radians)
double sin( double x);  // returns the sine of x (x in radians)
```

Ensure that any *C* code you write in answer to the questions below contains sufficient comments for the reader to understand the program.

SECTION A – Answer SIX parts of this section

1.1) Identify which of the following 10 names are *not* valid for *C* variables, giving a reason in each case.

```
7th            long
Canadian_$     longer
CHAR           twenty%
infinite       up&down
input_         zero
```

[7 marks]

1.2) The following three statements appear *before* the `main()` function of a *C* program is defined:

```
#include <stdio.h>
#define TWOPI 6.283185
int a=1;
```

Briefly describe the consequences of each statement.

[7 marks]

1.3) Inside a *C* function some variables are declared with the following statements:

```
float w = 0.5;
const float x = 0.5;
static float y = 0.5;
float *z = &w;
```

Distinguish clearly the consequences of these different declarations for each of the variables `w, x, y, z`.

[7 marks]

1.4) Briefly summarise the actions carried out by the following short *C* program, and describe the output that is produced.

```
#include <stdio.h>
int main()
{  float x[8];
   int i;
   for (i=0 ; i<8 ; i++)
   {  if (i%2 == 1) x[i] = -((float) i/10.0);
      else x[i] = (float) i/10.0;
      printf( "x[%d] = %5.2f \n", i, x[i]);
   }
   return 0;
}
```

[7 marks]

1.5) 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 is, in general, equivalent to a particular form of `for` loop.

[7 marks]

1.6) A structure of type `cylinder` is defined by the *C* code fragment below.

```
struct cylinder
{  char material[20];
   float density;     // in kg per cubic metre
   float length;      // in metres
   float radius;      // in metres
};
```

Provide a few additional lines of *C* code that will

(i) declare a variable named `rod` that is a structure of this type, initialised so that its members store the values `"copper"`, 8930, 0.2, 0.07, respectively,

(ii) use the member values to calculate the mass of the rod, and then print out 2 lines of information, in the format shown below.

```
Rod material: copper
Rod mass: 27.49 kg
```

[7 marks]

1.7) A *C* program initialises the variables `p, x, y` and `z` in the following way.

```
float x=8.0, y=12.0, z=5.0, *p;
p = &z;
```

What are the values of `x` after each of the *C* statements below have been executed?

```
(i)      x += y/z;
(ii)     x *=  y * (*p);
(iii)    x = (y<z) ? y : z;
```

[7 marks]

Note: Treat each part of this question independently, the statements are not consecutive steps in the same program.

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

```
char *UpperCase( const char *input);
float Average( const float data[ ], int num_points,
               float *min, float *max);
```

[7 marks]

SECTION B – Answer TWO questions

2) In a dynamics experiment, the position of a particle in two dimensions is measured repeatedly using polar coordinates (r, θ). An analysis of the particle's motion is required in terms of Cartesian coordinates. Write a *C* program that will allow the user to carry out the tasks listed below.

(i) Repeatedly prompt the user to enter a polar coordinate pair r, θ (with the two values entered on the same line, separated by a comma, and θ (theta) measured in degrees), until the user indicates the end of the data set by entering a negative value for the radial coordinate r.

[6 marks]

(ii) Convert each pair of input values to Cartesian coordinates x, y, and print these coordinates to the screen, accompanied by a suitable message, before prompting for the next pair of polar coordinates to be input.

[6 marks]

(iii) Determine the maximum, minimum, and mean values for both the x and y coordinates.

[8 marks]

(iv) Print out to the computer screen the total number of particle positions read into the program, followed by two lines of information giving the maximum, minimum and mean values of the Cartesian coordinates, one line for the x and one for the y coordinates.

[6 marks]

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

[4 marks]

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

$$f(x) = \frac{x}{\exp(x) - 1}$$

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

[10 marks]

Describe the **trapezoidal rule** for estimating the area under a curve, and list the main requirements of a *C* program that uses the trapezoidal rule for this purpose.

[10 marks]

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

$$\int_1^5 f(x)\, dx$$

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

[10 marks]

4) Describe the **bisection** method for locating a real root of the equation $f(x) = 0$ for x in the interval from x_0 to x_1, where $x_0 < x_1$, and $f(x)$ is a continuous real function, with $f(x_0) < 0$ and $f(x_1) > 0$.

[6 marks]

Explain briefly why the **Newton-Raphson** method is often used to improve the accuracy of a root that is found approximately by the bisection method, but is not recommended for use in the early stages of a root-finding algorithm.

[6 marks]

Determine a suitable interval within which you expect to find the single positive real root of the equation

$$x^3 - 6x^2 + 14x - 11 = 0$$

[2 marks]

Write a *C* program that uses the bisection method to find this root of the equation to an accuracy of 5 decimal places, and then prints out the value found.

[12 marks]

Estimate the number of iterations that will be required to find the root to this accuracy, based on the starting range of values you determined above.

[4 marks]

5) Describe the **straight insertion** method of sorting an array of N values into order. Straight insertion sorting is referred to as an N^2 method: state one consequence of this property.

[8 marks]

Write a *C* function named `SI_sort` to use the straight-insertion method to sort into ascending order N integer values that are held in an array.

[10 marks]

A set of experimental data is stored as formatted values in a plain text file called `data.txt`. The data consist of 500 integer values, stored as 500 lines of text, with one value on each line. Write a *C* program that will carry out the following tasks:

(i) Read the data from the specified file into a one-dimensional array named `input`.

(ii) Sort the data into ascending order using the function `SI_sort`.

(iii) Write out the sorted data to a file called `sortdata.txt`, using the same format as for `data.txt`.

[8 marks]

Comment briefly on any changes that you would consider making to your program if the file `data.txt` contained 5×10^4 values instead of 500.

[4 marks]