

Question 1.

Write a program which solves a quadratic equation of the form $ax^2 + bx + c = 0$ (the equation giving x as a function of a , b , and c is in the data book). Your program should include a function `findx` which should have a prototype of the form

```
int findx(double a, double b, double c, double *highx, double *lowx)
```

where a , b , and c are the coefficients and `highx` and `lowx` are the two possible values of x . The function should return 0 if the quadratic has been solved, 1 if $a=0$ (in which case the equation isn't a quadratic!), and 2 if the values of x are complex (you are NOT asked to write the program so as to find complex solutions). You will need to use the library function (in `math.h`)

```
double sqrt(double x);
```

Three example runs of the program are illustrated below:

```
Enter coefficients of quadratic equation of form ax^2+bx+c=0
Enter a : 1
      b : 2
      c : -3
x is either 1.000000 or -3.000000
```

```
Enter coefficients of quadratic equation of form ax^2+bx+c=0
Enter a : 0
      b : 2
      c : 4
x is -2.000000
```

```
Enter coefficients of quadratic equation of form ax^2+bx+c=0
Enter a : 1
      b : 1
      c : 1
Equation has complex solution
```

[30 marks]

Question 2.

Below is the main program of software which firstly reads in up to 1000 floating point numbers from a file (`numbers.dat`) and puts them in an array, and secondly finds the mean of the numbers and the lowest value.

```
#include <stdio.h>

// function prototypes
int read_data(int *, float *);
float find_mean(int, float *);
float find_low(int, float *);

int main(void)
{
    int number_data, fileok;
    float data_array[1000], mean, lowest_value;

    // read the data file
    fileok=read_data(&number_data, data_array);

    if(fileok==0){ // data have been read OK
        mean=find_mean(number_data, data_array);
        lowest_value=find_low(number_data, data_array);
        printf("Mean = %f\n", mean);
        printf("Lowest value = %f\n", lowest_value);
    }
}
```

An example data file is as follows:

```
10
4.8
7.2
3.6
5.7
3.9
1.1
2.9
1.4
1.2
9.8
```

The first number (an integer) indicates the number of floating point numbers which follow.

You are asked to write the code for the three functions `read_data`, `find_mean`, and `find_low` which read in the data from file, find the mean value and the lowest value respectively. You should ensure that possible file-handling errors are trapped appropriately and the user informed. You must not modify the main program.

[30 marks]

Question 3.

(a) Define the following C/C++ concepts and explain when or how they might be used:

(i) pointers

(ii) call by address

(iii) dynamic memory allocation

(iv) structs

[12 marks]

(b) The following program is designed to sort some names (held in an array of strings) into alphabetical order.

```
#include <stdio.h>
#include <string.h>

#define NUMNAMES 5;

int main(void) {

    int ii, jj;
    char test_names[NUMNAMES][30]={"Smith A.", "Jones P.",
        "Jackson Q.", "Smith C.", "Hamilton M."};
    char temp_name[30]

    for(ii=0; ii<NUMNAMES; ii++){
        for(jj=ii+1; jj<NUMNAMES; jj++){
            for(kk=0; kk<30; kk++){
                if(test_names[ii][kk]>test_names[jj][kk]){
                    strcpy(temp_name, test_names[ii]);
                    strcpy(test_names[ii], test_names[jj]);
                    strcpy(test_names[jj], temp_name);
                } else if (test_names[ii][kk]<test_names[jj][kk]) {
                    break; // from inner loop
                }
            }
        }
    }

    print("The names in alphabetical order are :\n\n");
    for(ii=0; ii<NUMNAMES; ii++){
        printf("%s\n", test_names[ii]);
    }
}
```

There are a number of errors in the above code, some syntactical (ie compile time) and some algorithmic (ie run time). Identify each error by numbering it on the working copy of the program (located at the end of the exam paper). You should also explain why you consider this to be an error and how you would correct it.

Remember to attach the working copy of the program to your examination answer book (and include you candidate number)

[18 marks]

```
#include <stdio.h>
#include <string.h>

#define NUMNAMES 5;

int main(void) {

    int ii, jj;
    char test_names[NUMNAMES][30]={"Smith A.", "Jones P.",
        "Jackson Q.", "Smith C.", "Hamilton M."};
    char temp_name[30]

    for(ii=0; ii<NUMNAMES; ii++){
        for(jj=ii+1; jj<NUMNAMES; jj++){
            for(kk=0; kk<30; kk++){
                if(test_names[ii][kk]>test_names[jj][kk]){
                    strcpy(temp_name, test_names[ii]);
                    strcpy(test_names[ii], test_names[jj]);
                    strcpy(test_names[kk], temp_name);
                } else if (test_names[ii][kk]<test_names[jj][kk]) {
                    break;    // from inner loop
                }
            }
        }
    }

    print("The names in alphabetical order are :\n\n");
    for(ii=0; ii<NUMNAMES; ii++){
        printf("%s\n", test_names[ii]);
    }
}
```