

W249/102

**DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8.**

BSc Information Systems / Information Technology

Stage 1

WINTER EXAMINATIONS 2007

PROGRAMMING AND ALGORITHMS

MR. C. O'LEARY
DR. B. O'SHEA

THURSDAY, 11TH JANUARY 2007, 9:30 TO 12:30

ATTEMPT 4 QUESTIONS

ALL QUESTIONS CARRY EQUAL MARKS

1. (a) Consider the incomplete code shown below:

```
#include <stdio.h>

int main() {

    char name[20];
    int age;
    float salary;

    printf("Enter your name  : ");
    scanf("%s", name);
    printf("Enter your age   : ");
    scanf("%d", age);
    printf("Enter your salary: ");
    scanf("%f", salary);

    printf("Name          : \n");
    printf("Age           : \n");
    printf("Salary        : \n");
}
```

The desired output of this program is shown below (text typed by the user is shown in bold).

```
Enter your name  : Tom Cruise
Enter your age   : 45
Enter your salary: 210000
Name           : Tom Cruise
Age            : 45
Salary         : 210000.00
```

Show how to change the above code so that the desired output is produced.

(6 marks)

- (b) With reference to the C programming language, answer the following 9 questions (1 mark each). Use sample code where possible to make your answer clear.

- i. Give an example of a *literal*.
- ii. Show how to perform a *type-cast*.
- iii. Give an example of an *invalid* variable name.
- iv. Why would you use an unsigned value rather than a signed value?
- v. What size is an `int` in C on the Win32 platform?
- vi. What size is a `double` in C on the Win32 platform?
- vii. What is the ASCII character set (don't give the entire set – just explain briefly what it is used for)?
- viii. What are the operators `||` and `&&` used for?
- ix. What are the two styles of comment that can be used in C?

(9 marks)

- (c) Describe clearly how *linked lists* are implemented.

Using *diagrams* and/or *pseudocode* and/or *C code* show how elements are:

- i. Inserted into the middle of a list.
- ii. Inserted at the end of a list.

(10 marks)

2. (a) Consider the incomplete code shown below:

```
#include <stdio.h>

int main() {

    int i;
    int limit;
    limit = argv[1];

    while(i = 0; i <= limit; i++) {
        if(i / 3 = 0) {
            printf("%d is divisible by 3\n", i);
        }
    }
}
```

This program is supposed to take a number (the *limit*) as a command line argument, and print out all the numbers divisible by 3 between 0 and this limit – as shown below:

```
c:\> q2a 20
0 is divisible by 3
3 is divisible by 3
6 is divisible by 3
9 is divisible by 3
12 is divisible by 3
15 is divisible by 3
18 is divisible by 3
```

Show how to change the code so that the desired output is produced.

(6 marks)

- (b) Consider the scenarios below where the user is asked to enter values until some condition is reached.

You are required to select the most appropriate type of loop, and provide the C code for the loop (you can use pseudocode for reading, writing etc). You must also briefly justify your choice of loop type.

- i. The user must select a menu item between 1 and 10. If, and only if, they enter an incorrect value, the loop should iterate another time.
- ii. The user must enter exactly 10 values.
- iii. The user must enter values until they elect to end the loop by entering a sentinel value.

(9 marks)

- (c) Describe in detail the operation and behaviour of *stack* and *queue* data structures. Show how *one of* these data structures could be implemented using an array, using either *pseudocode* or *C code*.

(10 marks)

3. (a) Consider the code shown below:

```
#include <stdio.h>

int main() {
    int month;
    char months[12][20] = {'January', 'February',
                          'March', 'April',
                          'May', 'June',
                          'July', 'August',
                          'September', 'October',
                          'November', 'December' };

    printf("Enter a month of the year (1 - 12): ");
    scanf("%d", &month);

    printf("You are currently in %s.\n", months[month]);
    printf("This is a ");

    switch(month) {
        case 0:
        case 10:
        case 11:
            printf("Winter");
        case 1:
        case 2:
        case 3:
            printf("Spring");
        case 4:
        case 5:
        case 6:
            printf("Summer");
        case 7:
        case 8:
        case 9:
            printf("Autumn");
    }
    printf(" month.\n");
}
```

This program is supposed to prompt the user for a month, between 1 and 12 (1 = January, 12 = December), and then print out both the month and the season (Winter = November, December, January; Spring = February, March, April; Summer = May, June, July; Autumn = August, September, October) – as shown below (text typed by the user is shown in bold):

```
Enter a month of the year (1 - 12): 5
You are currently in May.
This is a Summer month.
```

Show how to change the code so that the desired output is produced.

(6 marks)

- (b) With reference to arrays in the C programming language, answer the following 9 questions (1 mark each). Use sample code where possible to make your answer clear.

- i. What is the maximum size of an array?
- ii. What is the maximum number of dimensions for a multi-dimensional array?
- iii. How is the preprocessor used to declare the size of an array?
- iv. Why is the preprocessor sometimes used to declare the size of an array?
- v. What is the relationship between character strings and arrays?
- vi. What is the relationship between arrays and pointers?
- vii. Are arrays passed to functions by value or by reference?
- viii. How can memory be allocated at runtime for an array?
- ix. How can an array length be extended at runtime?

(9 marks)

- (c) *Sorting algorithms are divided into two classes based on their complexity.*

Identify both classes and describe clearly the characteristics of algorithms in each class.

Use *one* named algorithm from *each* class to sort the following list of numbers:

5 3 8 9 2 4 1 7

(10 marks)

4. (a) Consider the incomplete code shown below:

```
#include <stdio.h>

int main() {
    char *s = "Now is the winter of our discontent";
    printf("Length: %d\n", length_of_string());
}

void length_of_string(char *string) {
    int i;
    for(i = 0; string[i] != '\n'; i++) {
    }
}
```

This program is supposed to call the function `length_of_string`, which is supposed to calculate the length of the string passed as an argument and return the length as an `int`, which is then printed out.

Show how to change the program so that it prints out the length of the string declared as `s` in `main`. (Note, you may *not* use the `strlen` function.)

The desired output is shown below:

```
Length: 35
```

Show how to change the code so that the desired output is produced.

(6 marks)

- (b) Consider the situations below where a function is required to carry out an operation.

You are required to provide the C code for the function (you can use pseudocode for reading, writing etc) as well as the code required to call the function. You must also briefly justify your function design.

- i. `main` calls a function which takes an `int` as an argument. The function increments the value of the `int` passed as an argument. The new incremented value must be available to `main`.
- ii. `main` calls a function which takes two `ints` as arguments. The function then swaps the two numbers so that the first `int` is now in the memory location of the second `int`, and the second `int` is now in the memory location of the first `int`.
- iii. `main` needs to call a function to calculate the factorial of a number (multiplied by all numbers less than it and greater than 0). The function uses recursion to calculate the value and returns it to `main`.

(9 marks)

- (c) Identify *five* stages of the *Waterfall Lifecycle Model*, and describe in detail the processes that take place at each stage.

(10 marks)

5. (a) Consider the incomplete code shown below:

```
#include <stdio.h>

structure timetable_entry {
    int module_code;
    int number_of_students;
    char lecturer[20];
};

int main() {

    printf("Enter module code      : ");
    scanf("%d", module_code);
    printf("Enter number of students: ");
    scanf("%d", number_of_students);
    printf("Enter lecturer          : ");
    scanf("%s", lecturer);

    printf("Module code : %d\n", module_code);
    printf("Num students: %d\n", number_of_students);
    printf("Lecturer   : %s\n", lecturer);
}
```

This program is supposed to prompt the user to enter data relating to a timetable entry, store it all in a structure, and then print out the contents of the structure, as shown below (text typed by the user is shown in bold).

```
Enter module code      : 1234
Enter number of students: 34
Enter lecturer         : Ciaran
Module code : 1234
Num students: 34
Lecturer   : Ciaran
```

Show how to change the code so that the desired output is produced.

(6 marks)

- (b) Consider the situations below where a function is required to carry out an operation.

You are required to provide the C code for the function (you can use pseudocode for reading, writing etc) as well as the code required to call the function. You must also briefly justify your function design.

- i. `main` calls a function which takes no arguments but returns a `timetable_entry` (from part (a) above). The function prompts the user to enter the values for the structure.
- ii. `main` calls a function which takes a `timetable_entry` (from part (a) above) as an argument. The function prompts the user to enter new values for the structure, which then become available in the structure in `main`.
- iii. `main` calls a function which takes an array of `timetable_entry` (from part (a) above) structures and the size of the array as arguments. The function then calls the function from part ii above to fill the structures with new values, which then become available in the array in `main`.

(9 marks)

- (c) Discuss *open source* as a culture for software development.

(10 marks)

6. (a) Consider the incomplete code shown below:

```
#include <stdio.h>

int main() {

    FILE *file_pointer;
    file_pointer = fopen("a_file", "r");

    int file_size;

    // ADD CODE HERE

    printf("Size of file: %d\n", file_size);
}
```

This program is supposed to calculate the length of the file named `a_file` (you can assume that this file already exists).

The desired output of this program is shown below.

```
Size of file: 19
```

Add code to this program so that it correctly calculates the length of the file, and prints it out.

(6 marks)

- (b) With reference to file handling in the C programming language, answer the following 9 questions (1 mark each). Use sample code where possible to make your answer clear.

- i. What is a text file?
- ii. What is a binary file?
- iii. What is a file pointer?
- iv. What happens if a file is opened with permissions `r+`?
- v. What happens if a file is opened with permissions `w+`?
- vi. What file pointers are automatically opened for all processes?
- vii. How are characters written to, and read from, files?
- viii. How are strings written to, and read from, files?
- ix. How are bytes written to, and read from, files?

(9 marks)

- (c) Show, using diagrams, sample C code and pseudocode where necessary, how to create a simple database using structures and binary files. In particular, show how data can be written, overwritten and read.

(10 marks)