



DUBLIN INSTITUTE OF TECHNOLOGY

**DT211C BSc. (Honours) Degree in Computer Science
(Infrastructure)**

Year 1

SUMMER EXAMINATIONS 2015/2016

PROGRAMMING WITH PERSISTENT DATA [CMPU1028]

DR. PIERPAOLO DONDIO
DR. DEIRDRE LILLIS

MONDAY 23RD MAY 9.30 A.M. – 11.30 A.M.

DURATION

2 HOURS

INSTRUCTIONS TO CANDIDATES

ANSWER **THREE** QUESTIONS OUT OF **FOUR**.

ALL QUESTIONS CARRY EQUAL MARKS.

1 MARK IS ALLOCATED FOR SITTING THE PAPER

Question 1

[33 marks]

- a. Write a C program that takes the file *start.dat* as input, and it generates a file called *reverse.dat* containing all the data in the file *start.dat* in reverse order, i.e. the first byte of *start.dat* is the last byte of *reverse.dat*, the second byte of *start.dat* is the second last byte of *reverse.dat* and so on...

[14 marks]

start.dat (input)

1 st byte	2 nd byte	3 rd byte	n th - 2 byte	n th - 1 byte	n th byte
----------------------	----------------------	----------------------	-----	-----	--------------------------	--------------------------	----------------------

reverse.dat (output)

n th byte	n th - 1 byte	n th - 2 byte	3 rd byte	2 nd byte	1 st byte
----------------------	--------------------------	--------------------------	-----	-----	----------------------	----------------------	----------------------

- b. The above problem can be tackled with two strategies: reverse the entire file in memory or reverse the file by working directly with the disk.

- (i) Describe the solution that you did not implement in question a. (pseudocode or a detailed description in English is acceptable, C code is NOT required).

[9 marks]

- (ii) What are the differences and the advantages of each of the methods?

[4 marks]

- c. Describe the functioning of the binary search algorithm on a file, providing an example and explaining how to implement it in C. What is the key assumption on the file?

[6 marks]

Question 2

[33 marks]

- a. Consider the following C program:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 100

int main(int argc, char *argv[])
{
    FILE *fp;
    int count = 0;
    char filename[SIZE];
    char text[SIZE];
    printf("Please enter the name of the file you wish to
open: ");

    gets(filename);
    printf("Opening %s\n", filename);

    fp = fopen(filename, "r");

    if (fp == NULL)
    {
        printf("ERROR: File could not be found!");
        exit(1);
    }
    while (fgets(text, SIZE, fp) != NULL)
    {
        if (strstr(text, "exam") != NULL)
        {
            count++;
            printf("%s", text);
        }
    }
    printf("\nCount: %d", count);
    fclose(fp);
    return 0;}
```

- (i) What is the program doing? Comment the code and give an example of input/output
[6 marks]
- (ii) Is the program able to always count the occurrences of the word "exam" in the file? Justify your answer giving an example.
[4 marks]
- (iii) Is the program able to always count the number of lines containing the english word "exam"? Justify your answer giving an example.
[4 marks]

- b. Describe the necessary steps to create an index file for a large binary file of structures and for a specific key field. Provide examples of C instructions for each step of the process

[10 marks]

- c. Describe 3 disk scheduling algorithms. Compare them in terms of fairness and performance.

[3 marks each, 9 marks total]

Question 3

[33 marks]

- a. Write a C program to convert a text file containing 3 columns separated by a space character ' ' (ASCII code 32), where each column is an integer number (codified as text!) into a .csv file (a text file using comma as delimiter instead of the space character). An example of file:

```
23 34 5454
6 754 23
676 5 45
```

The output file will be:

```
23,34,5454
6,754,23
676,5,45
```

[15 marks]

- b. Write a program to convert a text file with the same format as the file at question a) into a binary file of integers.

[10 marks]

- c. Describe how to delete, update or insert data in a file using a flag to mark deleted or invalid data. What are the advantages?

[8 marks]

Question 4

[33 marks]

The file `result.txt` contains the following:

Mark	45	66
Mary	55	84
Paul	76	49
...		

Each line contains a name of a student (string of maximum size of 20), his/her CA marks and his/her Exam marks (both integer numbers from 0 to 100).

- (i) Write a C program to read the file and create three new files, `pass.txt`, `fail.txt` and `repeat.txt`. The file `pass.txt` contains the names of the students that passed the module. In order to pass the module, the average of CA and Exam marks have to be above or equal to 40 (note that a student might fail CA or EXAM, but if the average is above 40, the student passes the module). If a student has passed only one of the two components but the average is below 40, the name of the student is saved into `repeat.txt`. If the student failed both the exam and the CA his/her name is saved into `fail.txt`.

[15 marks]

- (ii) Modify the above program to display the name of the student that got the highest average marks.

[8 marks]

- (iii) What is a FAT? Describe its functioning, its content and advantages.

[10 marks]