

W249/102

DUBLIN INSTITUTE OF TECHNOLOGY  
KEVIN STREET, DUBLIN 8.

---

# **BSc Information Systems / Information Technology**

Stage 1

---

WINTER EXAMINATIONS 2009

---

**PROGRAMMING AND ALGORITHMS**

MR. C. O'LEARY  
DR. D. LILLIS

2 HOURS

ATTEMPT **3** QUESTIONS

ALL QUESTIONS CARRY EQUAL MARKS

1. (a) Provide the C code for a program which takes a string of text and a single character as command line arguments, and print out the number of times that the character appears in the text.

Your program must produce output in the form shown below:

```
C:\>q1 "Yes we can" e
e appears in "Yes we can" 2 times.
```

```
C:\>q1 "The end of the beginning" n
n appears in "The end of the beginning" 4 times.
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- What is meant by a type cast?
  - What is the main different between a while loop and a do..while loop?
  - What is a sentinel value?

**(6 marks)**

- (c) The code below is a partial implementation of a program to sort an array of numbers using QuickSort.

```
#include <stdio.h>

struct array_with_length {
    int length;
    int numbers[10];
};

int main() {

    array_with_length quicksort(array_with_length);
    int i;

    array_with_length a = {
        10,
        { 231, 82, 448, 67, 262, -357, -214, 166, -264, -264 }
    };

    for(i = 0; i < 10; i++) printf("%d ", a.numbers[i]);
    printf("\n");

    a = quicksort(a);

    for(i = 0; i < 10; i++) printf("%d ", a.numbers[i]);
    printf("\n");
}
```

Provide the code for the quicksort function to complete the implementation of the program.

**(12 marks)**

2. (a) Provide the C code for a program which finds the highest common factor of two integers provided by the user.

Your program must produce output in the form shown below (user input appears in bold):

```
C:\>q2
Enter the first number: 10
Enter the second number: 8
The highest common factor of 10 and 8 is 2

C:\>q2
Enter the first number: 99
Enter the second number: 66
The highest common factor of 99 and 66 is 33
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- What is the ASCII character set?
  - What is the difference between a text file and a binary file?
  - How is it possible to find the length of a file in a C program?

**(6 marks)**

- (c) The program below adds names and ages of students to a binary file.

```
#include <stdio.h>

struct student {
    char name[20];
    int age;
};

int main() {

    FILE *fp;
    student s;

    fp = fopen("./f.dat", "a");

    printf("Enter name: ");
    gets(s.name);
    printf("Enter age : ");
    scanf("%d", &s.age);

    fwrite(&s, sizeof(student), 1, fp);

    fclose(fp);
}
```

Provide a companion program that reads from the same file and prints out the names of all students whose ages have a value between 30 and 40.

**(12 marks)**

3. (a) Provide the C code for a program which replaces instances of multiple spaces in a line of text with a single space, and removes all leading spaces (spaces at the start of the text). The text is provided by the user.

Your program must produce output in the form shown below (user input appears in bold):

```
C:\>q3
Enter the text:      I   like  chips
Output: I like chips

C:\>q3
Enter the text: What's      your name?
Output: What's your name?
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- i. What is the relationship between arrays, strings and pointers?
  - ii. What are the advantages of using linked lists instead of arrays?
  - iii. What are the disadvantages of using linked lists instead of arrays?

**(6 marks)**

- (c) The code below provides a partial implementation of a queue in a C program which uses a linked list.

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

struct list {
    int data;
    list* next;
};

int main() {

    char choice;
    list* head;
    list* tail;

    head = tail = NULL;

    printf("Enter choice:\n");
    printf("\tPut (A)\n");
    printf("\tGet (B)\n");
    printf("\tExit (X)\n");
    printf("\t\t--> ");
    choice = (char)getchar();
    getchar();
    while(choice != 'X') {
        switch(choice) {
            case 'A':
            case 'B':
        }
        printf("Enter choice:\n");
        printf("\tPut (A)\n");
        printf("\tGet (B)\n");
        printf("\tExit (X)\n");
        printf("\t\t--> ");
        choice = (char)getchar();
        getchar();
    }
}
```

Provide the remaining code for this program by rewriting the highlighted switch statement and adding additional variables if necessary, such that the behaviour of the program is that of a queue.

Note: You do not need to rewrite all the above code, simply provide the necessary additions / amendments.

**(12 marks)**

4. (a) Provide the C code for a program which takes two strings of text as command line arguments, and prints out whether the second string is a substring of the first string, and if so, the position at which it starts.

Your program must produce output in the form shown below:

```
c:\>q4 Oligopoly gop
"Oligopoly" contains "gop" at position 3
```

```
c:\>q4 Oligopoly pog
"Oligopoly" does not contain "pog"
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- i. What is the role of the pre-processor in C programming?
  - ii. What is the different between argument passing by reference and by value?
  - iii. What is a function signature?

**(6 marks)**

- (c) The code below provides a partial implementation of a stack in a C program which uses dynamic memory allocation to extend the size of the stack when necessary. The stack uses an array rather than a linked list, with the size of the array growing with the size of the stack.

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

int main() {

    int* stack;
    int size_of_stack;
    char choice;

    size_of_stack = 0;

    printf("Enter choice:\n");
    printf("\tPush (A)\n");
    printf("\tPop (B)\n");
    printf("\tExit (X)\n");
    printf("\t\t--> ");
    choice = (char)getchar();
    while(choice != 'X') {
        switch(choice) {
            case 'A':
            case 'B':
        }
        printf("Enter choice:\n");
        printf("\tPush (A)\n");
        printf("\tPop (B)\n");
        printf("\tExit (X)\n");
        printf("\t\t--> ");
        choice = (char)getchar();
    }
}
```

Provide the remaining code for this program by rewriting the highlighted switch statement and adding additional variables if necessary, such that the behaviour of the program is that of a stack.

Note: You do not need to rewrite all the above code, simply provide the necessary additions / amendments.

**(12 marks)**