

RW249/102

DUBLIN INSTITUTE OF TECHNOLOGY  
KEVIN STREET, DUBLIN 8.

---

# **BSc Information Systems / Information Technology**

**Stage 1**

---

**SUPPLEMENTAL 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 that takes a string of text as an argument and checks its validity as a password, where all passwords must:
- i. Be between 8 and 12 characters long
  - ii. Contain at least 3 lowercase letters (a-z)
  - iii. Contain at least 3 uppercase letters (A-Z)
  - iv. Contain at least 2 digits (0-9)

Note: you may not use `strlen` to calculate the length of the string.

Your program must produce output in the form shown below:

```
C:\>q1 mysecret
"mysecret" is an invalid password.

C:\>q1 mySEcRet01
"mySEcRet01" is a valid password.

C:\>q1 mySEcRet011qwerty
"mySEcRet011qwerty" is an invalid password.
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- i. What are the rules for valid variable names?
  - ii. What is the relationship between arrays and strings?
  - iii. How are multi-dimensional arrays created?

**(6 marks)**

- (c) The code below provides a partial implementation of a linked list in C. Items are added to the end of the list, and can be printed out.

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

int main() {

    char choice;
    list* head, tail, l;

    printf("Enter choice:\n");
    printf("\tAppend (A)\n");
    printf("\tPrint (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("\tAppend (A)\n");
        printf("\tPrint (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 linked list with items appended to the end.

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

**(12 marks)**

2. (a) Provide the C code for a program that takes a string of text as an argument and outputs the text encrypted using a Caesar Cipher – whereby all characters are replaced by the character 3 positions further on in the alphabet. Uppercase letters are always replaced with uppercase letters, and lowercase letters are always replaced with lowercase letters.

For example, all 'A' characters are replaced by 'D', 'B' is replaced by 'E', 'a' is replaced by 'd', 'z' is replaced by 'c' and so on. Space characters, digits and punctuation do not need to be replaced.

State any assumptions you make.

Your program must produce output in the form shown below:

```
C:\>q2 "Happy days are here again"
Plaintext:  Happy days are here again
Ciphertext: Kdssb gdbv duh khuh djdlq
```

```
C:\>q2 "Ireland 1-1 Holland"
Plaintext:  Ireland 1-1 Holland
Ciphertext: Luhodqg 1-1 Kroodqg
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- i. What is the ASCII character set?
  - ii. What is the difference between a text file and a binary file?
  - iii. How is it possible to perform random access on a binary file?

**(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)**

3. (a) Provide the C code for a program that takes a number as an argument and outputs the multiplication tables for all values up to that number.

Your program must produce output in the form shown below:

C:\>q3 3

```

      1  2  3
-----
1 |  1  2  3
2 |  2  4  6
3 |  3  6  9

```

C:\>q3 9

```

      1  2  3  4  5  6  7  8  9
-----
1 |  1  2  3  4  5  6  7  8  9
2 |  2  4  6  8 10 12 14 16 18
3 |  3  6  9 12 15 18 21 24 27
4 |  4  8 12 16 20 24 28 32 36
5 |  5 10 15 20 25 30 35 40 45
6 |  6 12 18 24 30 36 42 48 54
7 |  7 14 21 28 35 42 49 56 63
8 |  8 16 24 32 40 48 56 64 72
9 |  9 18 27 36 45 54 63 72 81

```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:

- i. What is the signature of a function?
- ii. What is meant by *passing by value* to a function?
- iii. What is meant by *passing by reference* to a function?

**(6 marks)**

- (c) Devise and clearly describe using pseudocode or C code a method for maintaining a linked list in persistent storage (binary file). Consider all situations including addition of a new element at the start, in the middle and at the end of the list, as well as deletion of an element from the list.

**(12 marks)**

4. (a) Provide the C code for a program that takes a number representing a year as an argument and outputs whether the year is a leap year or not. A year is a leap year if it is divisible by 4 and not divisible by 100, unless it is divisible by 400.

2004 was a leap year because it was divisible by 4 but not by 100.

2000 was a leap year because it was divisible by 4 and 400.

1900 was not a leap year because it was divisible by 100 and not by 400

Your program must produce output in the form shown below:

```
C:\>q4 2004
2004 is a leap year

C:\>q4 2000
2000 is a leap year

C:\>q4 1900
1900 is not a leap year
```

**(15 marks)**

- (b) Answer the following short questions, using example code where necessary:
- What is the role of the *pre-processor* in C programming?
  - What is the difference between a *static* and *auto* variable in a function?
  - Why is the use of *global variables* not advisable in C programs?

**(6 marks)**

- (c) Describe clearly the operation of the *quicksort* algorithm, and use it to sort the following numbers:

51 78 68 10 37 25 87 82 28

**(12 marks)**