GX1334

OLLSCOIL na hEIREANN
THE NATIONAL UNIVERSITY of IRELAND

NATIONAL UNIVERSITY OF IRELAND, GALWAY

SUMMER EXAMINATIONS 2000

Second University Examination in Information Technology
Second University Examination in Electronic Engineering and Computing

CT229 PROGRAMMING II

Prof. D. Bell
Dr. G. Lyons
Dr. S. Redfern
Mr. C. O' Riordan

Time allowed: **Three hours**

Answer 2 questions from each section
All questions carry equal marks
Use separate answer books for each section

SECTION A

Q.1. (i) Examine the following program, and write down the exact output that it would give:

```
#include <stdio.h>

main() {
     int a= 6, b= 7, *c, *d;

     c= &a;
     d= &b;
     printf("%d %d\n", a, *d);
     b= *c*b;
     *c= a+*d;
     printf("%d %d\n", *c, b);
     }
```

(ii) Write a program that dynamically creates, and then destroys, an array of integers of size determined by the user.

Q.2. (i) What does the sizeof operator do? What are the following 2 functions used for?:

```
void* malloc(size_t size);
void* calloc(size_t nitems, size_t size);
```

(ii) Why is the `free(void*)` function essential when using `calloc` and `malloc`?

(iii) Write a function that finds the longest common prefix of two words that it receives as arguments, and returns this to the calling routine. For example, the longest common prefix of "global" and "glossary" is "glo", of "department" and "depart" is "depart", and of "glove" and "dove" is the empty string.

Q. 3. (i) Examine the following program, and write down the exact output that it would give:

```
#include <stdio.h>

int main(int argc, char *argv[])
     {
     int i;

     printf("\n argc = %d\n\n", argc);

     for (i = 0; i < argc; ++i)
     printf(" arg %d : %s\n", i, argv[i]);

     return 0;
     }
```

(ii) Write a command-line driven program that takes two parameters: filename and search string. The program should open the text file that the user requests (if the file exists) and determine how many times (if any) the search string is found in that file.

Q.4. (i) In a particular software system, the following structure has been defined, for storing customer information in a singly-linked list:

```
struct node
     {
     char company_name[30];
     char contact_name[40];
     char address[50];
     struct node* next;
     };

typedef struct node node;
```

The system maintains two pointers: one to to a "current" node, and the other to the "head" node in the list. The last node ("tail") is identified by its NULL valued next member.

Write the code necessary to perform the following functions:

```
void InsertAfter(node* head, node* current);
// inserts a new node after the one pointed to by
// current, and sets current to point to this new
// node

int GetLength(node* head);
// return a count of the number of nodes currently
// in the system

void MovePrevious(node* current);
// resets current to point to the previous node
```

SECTION B

Q.5. (i) Compare arrays, linked lists and binary search trees as representations of a sorted list.

(ii) Discuss, with the aid of an example, a technique that may be used to guarantee a balanced binary search tree.

(iii) Provide C code fragments to implement an insertion into a binary search tree (you may assume numeric values).

Q.6. (i) Compare the efficiency of bubble sort, insertion sort and selection sort.

(ii) Merge-sort and quicksort are two examples of divide and conquer approaches to sorting. Compare the two algorithms.
Provide C code fragments to illustrate how you would implement either of the above algorithms.

(iii) A student file consists of records containing the following student information: name, address, date of birth, and exam result. The file is to be sorted on exam results (a score value in the range 0 to 100). Outline an efficient algorithm to perform this sort.

Q.7 (i) Describe the technique of Huffman encoding. Illustrate the technique on the following string:

"to be or not to be"

(ii) Outline, with code fragments, possible representations of the graph data structure.

(iii) Compare and contrast the depth-first and breadth first traversal algorithms of a graph data structure. Provide pseudocode.