NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF APPLIED SCIENCE
COMPUTER SCIENCE DEPARTMENT
DECEMBER 2002 EXAMINATIONS

SUBJECT: STRUCTURED PROGRAMMING
CODE: SCS 2107

Instructions to candidate:

Answer any *FOUR* (4) questions
All questions carry equal marks [25 each]

3 HOURS

QUESTION ONE

a) Give advantages of structured programming. [4]

b) What are the 3 main constructs of structured programming? [3]

c) Explain any five desirable features of "good code". [5]

d) What do you understand by the term "good design". [4]

e) Write a function that uses the bubble sort to sort an array of integers. Your function should accept arguments by reference and NOT by value. [9]

QUESTION TWO

a) What do we mean by the ``equivalence between arrays and pointers'' in C? [3]

b) If p is a pointer, what does p[i] mean? [3]

c) What's wrong with this #define line?

```
#define N 10;
```
[2]

d) If the header file x.h contains an external prototype declaration for a function q(), where should x.h be included? [3]

e) The assignment in

```
char c;
int *ip = &c;          /* WRONG */
```

is in error; you can't mix char pointers and int pointers like this. How, then, is it possible to write

```
char *cp = malloc(10);
int *ip = malloc(sizeof(int));
```

without error on either line? [4]

f) Write a program which presents a menu and requests an input character. When the user enters the character, the type associated with that character is displayed on the screen. The **c** in the following sample screen output was entered by the user. This caused the program to display char followed by a termination message.

```
Screen output:
Enter a character as defined below:
(i) for int
(f) for float
(c) for char
(d) for double, or
(L) for long
```

[10]

QUESTION THREE

a) Write the function

```
int countchars(char string[], int ch);
```

which returns the number of times the character ch appears in the string. For example, the call

```
countchars("Hello, world!", 'o')
```

would return 2. [10]

b) Write a short program to read two lines of text, and concatenate them using strcat. Since strcat concatenates in-place, you will have to make sure you have enough memory to hold the concatenated copy. For now, use a char array which is twice as big as either of the arrays you use for reading the two lines. Use strcpy to copy the first string to the destination array, and strcat to append the second one. [15]

QUESTION FOUR

a) By use of struct arrays that (i) accept student info (student id, surname, name, and year (numerical) of study), write a program with the following functions,

add, to add a new student if the array is not full
search, to search for a student in the array (by student id)
delete, to erase a student from the array
display, to display a student's info (note search, delete should use this function)

Your functions should be of void type hence parameter passing is essentially by reference [20]

b) What modifications would you make to your program in (a) above to store your data more permanently (i.e.in a file)? [5]

QUESTION FIVE

Write a ***structured*** C program with the following arithmetic functions:

"1. Add";
"2. Subtract";
"3. Multiply";
"4. Divide";
"5. Power";
"6. Square root";
"0. Exit";

[Note except for option 0, each option should be a function in its own right] [15]

c) Write two functions, one that reads into arrays and the other outputs the array contents, via pointers. [10]

QUESTION SIX

Write a rudimentary checkbook balancing program. It will use getline to read a line, which will contain either the word "check" or "deposit". The next line will contain the amount of the check or deposit. After reading each pair of lines, the program should compute and print the new balance. You can declare the variable to hold the running balance to be type float, and you can use the function atof (also in the standard library) to convert an amount string read by getline into a floating-point number. When the program reaches end-of-file while reading "check" or "deposit", it should exit. (In outline, the program will be somewhat similar to the average-finding program.)

For example, given the input

```
deposit
100
check
12.34
check
49.00
deposit
7.01
```

the program should print something like

```
balance: 100.00
balance: 87.66
balance: 38.66
balance: 45.67
```

[25]

END OF QUESTION PAPER

GOOD LUCK!