UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

PROGRAMMING TECHNIQUES I

COURSE CODE – EE271

MAIN EXAMINATION

DECEMBER 2011

DURATION OF THE EXAMINATION - 3 HOURS

INSTRUCTIONS TO CANDIDATES

1. There are FIVE questions in this paper. Answer any FOUR questions only.
2. Each question carries equal marks.
3. Use correct notation and show all your steps clearly in any program analysis.
4. All programs should be well documented and indented.
5. Start each new question on a fresh page.

Question 1

a)

i. Explain the rationale for using iterative control structures in C programming. [1]

ii. Give three examples of iterative control structures. For each control structure given as an example, explain its semantics. [11]

b) Explain the difference between *semantic* and *syntactic* programming errors and give one example of each. [3]

c) Explain the following terms as used in programming languages: [3]

(i) Machine Language
(ii) Assembly Language
(iii) High Level Language

d) Explain the following terms in the C programming language:

(i) Variable [4]
(ii) Variable Scope [3]

Question 2

Assume that reading is from the keyboard and output is displayed on the screen. Also assume that the following declarations have already been given.

```
struct product {
     char name[30];
     int price;
} inventory[5];
```

Write executable C statements with proper syntax (not a complete program), to perform each of the following tasks independently. Use only the declarations given above.

(i) Enable the user to capture a list of five products and their values into an inventory list. [2]

(ii) Display the matching range of each product according to the following criteria: If the price of the product is E10-E50 display "LOW COST"; if the value is greater that E50 but less than or equal to E150, display "MEDIUM COST"; if the value is greater than E150 but less than or equal to E400, display "HIGH COST"; otherwise display "UNKNOWN PRODUCT COST". [5]

(iii) Display the most expensive product in the list. [5]

(iv) A function to sort the product inventory list in ascending order according to product prices. [8]

(v) Assuming a inventory list of 50 products, display all the products in the 'LOW COST' and 'HIGH COST' categories according to the criteria in (ii). [5]

Question 3

A company that wants to send data over the internet has requested you to write *two* C programs.

The *first* program encrypts the data so that it may be transmitted more securely and reduce the risk of unauthorised users reading it. All data is transmitted as *four-digit positive* integers. Your program should read a four-digit positive integer entered by the user, encrypt and print the number using the following *encryption scheme*:

> *Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth.*

The *second* program inputs an *encrypted* four-digit integer and *decrypts* it (by reversing the encryption scheme above) to recover the *original* integer.

Your programs will be graded according to the following criteria:

i. *Correctness* – does the program produce the desired result i.e. for a specified number of lines and printing direction, can the program produce the correct Pascal triangle. [20]
ii. *Clarity* - proper indentation of program makes it easy to read. [1]
iii. *Sensible naming of variables* – make it easy to understand code when debugging. [2]
iv. *Proper use of comments* – comments also make the program easy to understand. [2]

Question 4

(a) Using **only** *recursive* functions (no repetition statements), write a program that displays the following checkerboard pattern: [12]

```
******************
********  ********
*******    *******
******      ******
*****        *****
****          ****
***            ***
**              **
*                *
**              **
***            ***
****          ****
*****        *****
******      ******
*******    *******
********  ********
******************
```

Your program must use **only** three output statements, one of each of the following forms:

```
printf("* ");
printf(" ");
printf("\n");
```

(b) Analyse the following program and determine its output. [13]

```
#include <stdio.h>

int main(void) {
```

```
        int r = 1;
        int s;

        while (r <= 8) {
                printf("%s", r%2 ? "" : " ");
                for(s=1; s<=8; s++){
                        printf("* ");
                }
                printf("\n");
                r++;
        }
        return 0;
}
```

Question 5

Carefully analyse the program shown in Figure 5 and determine its output. Show all working. [25]

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

/* Function prototypes */
void function1();
void function2();
void function3();
void function4();

/* Global variables  */
int M[5][5] = {0};
int r;
int c;

int main (void){
        function1();
        function2();
        function3();
        function4();
        printf("Press any key to continue.....\n");
        while(!kbhit()) { }
        return 0;
}
void function1() {
        for(r=0; r<5; r++) {
                for(c=0; c<5; c++) {
                        M[r][c] = 2*r*r + 3*r*c + 4*c*c;
                }
        }
}
void function2() {
        r=0;
        while(r<5) {
                for(c=0; c<5; c++) {
                        if(M[r][c]%2==0) {
                                M[r][c] = sqrt(M[r][c]/2);
                        } else {
                                M[r][c] = pow(3,M[r][c]%2);
                        }
                }
                r++;
        }
}
void function3() {
        for(r=0; r<5; r++) {
                for(c=0; c<5; c++) {
                        if(r==c) {
                                M[r][c] = M[r][c]*r*c;
                        }
                }
        }
}
void function4() {
        for(r=0; r<5; r++) {
                for(c=0; c<5; c++) {
                        if(r>=c) {
                                printf("%d ", M[r][c]);
                        } else {
                                printf(" ");
                        }
                }
                printf("\n");
        }
        for(r=3; r>=0; r--) {
                for(c=0; c<5; c++) {
                        if(r>=c) {
                                printf("%d ", M[r][c]);
                        } else {
                                printf(" ");
                        }
                }
                printf("\n");
        }
}
```

Figure 5. Program for Question 5

END OF EXAM PAPER