UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE & ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

PROGRAMMING TECHNIQUES I

COURSE CODE – EE271

MAIN EXAMINATION

DECEMBER 2012

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 for clarity.
5. Start each new question on a fresh page.

DO NOT OPEN THIS PAPER UNTIL PERMISSION HAS BEEN GIVEN BY THE CHIEF INVIGILATOR.

Question 1

a) Explain the differences and relationships between a *Psuedocode*, an *Algorithm* and a *Program.* [6]

b) Explain why it is always advisable to start by formulating a Psuedocode before embarking on writing a program. [3]

c) Compare and contrast the concepts of *recursion* and *iteration.* [5]

d)

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

ii. Give *three* examples of repetition control structures. For each of the repetition structures given as examples, explain their semantics. [8]

e) Using an example, compare and contrast *counter-controlled* looping and *sentinel-controlled* looping. [2]

Question 2

(a) Find and correct the errors in each of the following program segments:
Assume:

```
int *zPtr;
int *aPtr = NULL;
void *sPtr = NULL;
int number, i;
int z[5] = {1, 2, 3, 4, 5};
sPtr = z;
```

(i) `++zPtr;` [3]

(ii) `number = zPtr;` [3]

(iii) `number = *zPtr[2];` [3]

(iv) [3]

```
for(i=0; i<=5; i++) {
     printf("%d", zPtr[i]);
}
```

(v) `number = *sPtr;` [4]

(vi) `++z;` [3]

(b) A palindrome is a number or a text that reads the same backward and forward. For example, each of the following five-digit integers is a palindrome: 12321, 45554, and 11611. Write a function that reads a five-digit string of alphanumeric characters and determines whether or not it is a palindrome. Your function should return a 1 if the string is a palindrome and 0 otherwise. [6]

Question 3

Using **only** *recursive* functions (NO repetition statements), write a program that displays the following checkerboard pattern: [25]

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

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

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

Question 4

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

```
#include <stdio.h>

void f(int a, int b, int c);

int main(void) {
      int n=1;
      int m=1;
      int max=10;

      while(n<=max) {
            f(n,m,max);
            n++;
            printf("\n");
      }
      return(0);
}

void f(int a, int b, int c) {
      while(b<=c) {
            if((a*b < c+5)||(a*b>c*5)||((a*b%2==0)&&(a*b>c*3))) {
                  printf("* ");
            } else {
                  ((a/b)%2==1)? printf("$ "):printf("# ");
            }
            b++;
      }
}
```

Question 5

Write a C program that takes as input a sequence of single digit *positive* integers, one integer at a time. Your program should continuously examine the input sequence. If sequence matches a specified 4-digit sequence of numbers (e.g. "2020") then the program should print the message: "***SEQUENCE DETECTED!***". The program should continue to accept the sequence of numbers until the user enters the decimal integer 100. This will signal that the user wishes to terminate the program. ***Your source code should NOT exceed 40 lines.***

Your programs will be graded according to the following criteria:

i. *Correctness* – does the program produce the desired result i.e. for a specified 4-digits sequence, is the program able to detect the special sequence from the input sequence. [22]

ii. *Clarity* - proper indentation of program makes it easy to read. [1]

iii. *Proper use of comments* – comments also make the program easy to understand. [2]

END OF EXAM PAPER