UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

PROGRAMMING TECHNIQUES I

COURSE CODE – EE271

MAIN EXAMINATION DECEMBER 2010

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.

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

Question 1

a) Explain the difference between *semantic* and *syntactic* programming errors. Give one example of each. [4]

b) Explain the following terms as described in programming languages:

i. Machine language
ii. Assembly language
iii. High-level language [3]

c) Using an example, explain the general syntax of the following control structures:
i. *While* loop [3]
ii. *For* loop [3]

d) Using examples, compare and contrast a while loop and a do while loop. [4]

e) Explain the term *Psuedocode* as used in computer programming. Why is it advisable to start by developing a Psuedocode before implementing the actual program? [3]

f) Explain the following terms as used in the C programming language:
(i) Variable [3]
(ii) Variable Scope [2]

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 products {
     char name[15];
     int price;
     int quantity;
} products[10];
```

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 ten products. For each product capture the name, price, and quantity. [2]

(ii) Display a list of products that cost a given price. If no match is found the program should display an appropriate message [5]

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

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

(v) Assuming a list of 100 products, write a *recursive* function to display all the products in the list. [5]

Question 3

Answer each of the following. Assume that single-precision floating point numbers are stored in 4 bytes, and that the starting address of the array is at location 1002500 in memory. Each part of this question should use the results of previous parts where appropriate.

a. Define an array of type float called *numbers* with 10 elements, and initialize the elements to the values 0.0, 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9. Assume the symbolic constant SIZE has been defined as 10. [2]
b. Define a pointer, *nPtr*, that points to an object of type float. [1]
c. Print the elements of array numbers. Print each number with 1 position of precision to the right of the decimal point. [4]
d. Give two separate statements that assign the starting address of array *numbers* to the pointer variable *nPtr*. [3]
e. Print the elements of array numbers using pointer/offset notation with the pointer nPtr. [5]
f. Assuming that *nPtr* points to the beginning of array numbers, what address is referenced by *nPtr + 8*? What value is stored at that location? [3]
g. Assumming that nPtr points to numbers[5], what address is referenced by nPtr -= 4. What is the value stored at that location. [7]

Question 4

(i) Carefully analyse the program shown in Figure Q4 and determine its output. Show all working clearly. [20]

(ii) *Identify* and *explain* **ten** syntactic errors from the C program shown in Figure Q5. For each error specify the line at which it has been identified. For example:

Line 4: Missing semicolon at the end of statement. Every C statement should be terminated with a semicolon.

[5]

Question 5

Write a C program to compute and print a Pascal triangle. Your program should take as input the number of lines to be computed and printed *n*. Your program should give the user four options for printing the triangle: *left*, *right*, *up*, and *down*. For example with the *down* option selected for printing and $n = 5$ lines, the output should be as shown below. Your program should be able to compute and print up-to 15 rows of the triangle.

```
1  4  6  4  1
 1  3  3  1
  1  2  1
   1  1
    1
```

Your program 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. [15]

ii. *Clarity* - proper indentation of program makes it easy to read. [2]

iii. *Sensible naming of variables* – make it easy to understand code when debugging. [3]

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

v. *Program modularity*- has the programmer made good use of functions for performing specific tasks in the program? [3]

```
#include <stdio.h>

int n = 9;
int x;
int t;
int s;


int main (void){

	for(x=0; x<n; x++) {
		if(x<=(n-5)) {
			for(t=0; t<(x+1); t++){
				printf("*");
			}
			for(s=0; s<(5-(x+1)); s++) {
				printf(" ");
			}
			printf(" ");
			for(s=0; s<(5-(x+1)); s++) {
				printf(" ");
			}
			for(t=0; t<(x+1); t++){
				printf("*");
			}
		}
		if(x>(n-5)) {
			for(t=0; t<(n-x); t++){
				printf("*");
			}
			for(s=0; s<(5-(n-x)); s++) {
				printf(" ");
			}
			printf(" ");
			for(s=0; s<(5-(n-x)); s++) {
				printf(" ");
			}
			for(t=0; t<(n-x); t++){
				printf("*");
			}
		}
		printf("\n");
	}
	return 0;
}
```

Figure Q4. Program for Question 4 (i)

```
1 #include <stdio.h>
2
3
4 int number1
5
6 int Main(void)
7       int 2s == 3;
8
9       printf("\nEnter number: );
10      Scanf("%i", &2s)
11
12      Printf("The Number is: %i", number));
13
16      return "executed successfully";
17
18 }
19
20
```

Figure Q5. Program for Question 4 (ii)

END OF EXAMINATION PAPER