UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE & ENGINEERING

DEPARTMENT OF ELECTRICAL &ELECTRONIC ENGINEERING

RESIT EXAMINATION

JULY 2018

TITLE OF PAPER: PROGRAMMING TECHNIQUES I

COURSE CODE: EEE271

DURATION: 3 HOURS

INSTRUCTIONS:

1. There are five (5) questions in this paper. Answer question 1 and any other three (3) questions.
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 sufficiently commented and indented for clarity.
5. Start each question in a new page.

This paper should not be opened until permission has been given by the invigilator.

This paper contains five (5) pages including this page.

Question 1. [25 Marks]

a. In general, why are iterative control structures used in programming? [2]

b. Give three examples of iterative control structures. For each control structure give an example. [9]

c. What is the difference between a break and a continue statement? [4]

d. What is recursion? Give an example of a recursive function. [5]

e. Define the following in the context of the C programming language:

i. Variable [1]

ii. Variable scope [2]

iii. Array [2]

Question 2 [25 Marks]

Given that single-precision floating point numbers are stored in 4 bytes of memory, and that the first element of an array is located at address 1002500 in memory. Do the following. Use answers from the previous parts of the question where appropriate.

a. Define an array called **numbers** 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**, which points to an object of type float. [1]

c. Print the elements of array **numbers**. Print each value 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. 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]

f. Print the elements of array **numbers** using pointer offset notation with the pointer **nPtr**. [5]

g. Assuming that nPtr points to **numbers**[5], what address is referenced by **nPtr** += 4. What is the value stored at that location. [7]

Question 3 [25 Marks]

a. Write a function *reverse* that takes as its arguments the following:
(1) an array of floating point values;
(2) an integer that tells how many floating point values are in the array.

The function must reverse the order of the values in the array. For instance, if the array passed to the function is:

0	1	2	3	4
8.8	2.6	9.0	3.4	7.1

Then when the function returns, the array will have been modified so that it is:

0	1	2	3	4
7.1	3.4	9.0	2.6	8.8

The function should not return any value.

[10]

b. For the following questions assume that reading of values is from the keyboard and output is displayed on the standard output. Further, assume that the following structure declaration has already been made.

```
struct products{
        char name[15];
        int price;
        int quantity;
        }product[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. [3]

ii. Display a list of products that cost a given price. If no match is found the program should display an appropriate message. [6]

iii. Display the cheapest product in the list. [6]

Question 4 [25 Marks]

Write a function called deleteRepeats that takes in a partialy filled array and deletes all repeated letters from the array. Since a partially filled array requires two arguments, the function will actually have two parameters: an array parameter and a parameter of type int that gives the number of array positions used. When a letter is deleted, the remaining letters are moved forward to fill in the gap. This will create empty positions at the end of the array so that less of the array is used. The second parameter of type int must tell how many array positions are filled. This second parameter will be a call-by-reference parameter and will be changed to show how much of the array is used after the repeated letters are deleted. For example, consider the following program segment:

```
char a[10];
a[0] = 'a';
a[1] = 'b';
a[2] = 'a';
a[3] = 'c';
int size = 4;
deleteRepeats(a, size);
```

After this program is executed, the value of a[0] is 'a', the value of a[1] is 'b', the value of a[2] is 'c', and the value of size is 3. (The value of a[3] is no longer of any concern, since the partially filled array no longer uses this indexed variable.)

Note:

- You may assume that the partially filled array contains only lowercase letters.
- Embed your function in a suitable test program.

Your program will be graded according to the following criteria:

i.	Correctness – does the program produce the desired result?	[18]
ii.	Clarity – proper indentation of program makes it easy to read.	[3]
iii.	Sensible naming of variables – for documentation and debugging purposes	[2]
iv.	Proper use of comments – also make the program easy to read.	[2]

Question 5 [25 Marks]

Write a program that uses a function called *diamond()* to displays a checkerboard pattern that can be zoomed in and out by 1 level. A normal sized diamond is 5 by 5 asterisks. A sample output is shown in Figure 5.1:

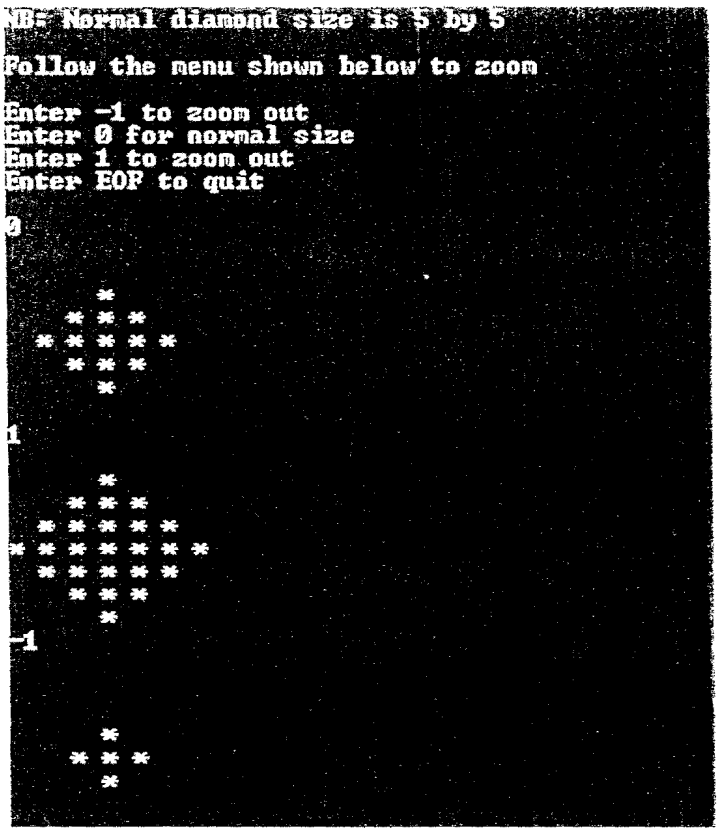

Figure 5.1

Your program must use repetition statements with a combination of the following output statements:

```
printf("  ");

printf "*" ;

printf " " ;

printf "\n" ;
```

Your program will be graded according to the following criteria:

i.	Correctness – does the program produce the desired result?	[18]
ii.	Clarity – proper indentation of program makes it easy to read.	[3]
iii.	Sensible naming of variables – for documentation and debugging purposes	[2]
iv.	Proper use of comments – also make the program easy to read.	[2]

End of Paper