**Francis Wilfred B. Olilang**
**CMSC 21 - Second Long Examination Answer Sheet**
**May 22, 2022**

**Part I.**

1. True
2. True
3. False
4. False
5. False
6. True
7. False
8. False
9. True
10. True
11. False

**Part II.**

1. The first dimension in a multidimensional array parameter can be left unspecified but not the other dimensions as the compiler can deduce (or even ignore) the first dimension from the context initialization. Moreover, the other dimensions must be there to tell to the compiler that the array contains multiple parameters.
2. 

```
a) bool isPalindrome(char string[],
                int i, int j)
{
    if (i == j)
    return true;

    if (string[i] != str[j])
    return false;

    if (i < j + 1)
    return isPalindrome(string, i + 1, j - 1);

    return true;
}
```

```
b) double computeAverage(int arr[20], int n)

{

     int sum = 0;

     for (int i=0; i<n; i++)

          sum += a[i];


     return (float)sum/n;

}
```

```
c) void reverseSentence (void) {}
```

d)
```
int squareRoot(int num)
{
    if (num == 0 || num == 1)
    return num;

    int i = 1, result = 1;
    while (result <= num)
    {
      i++;
      result = i * i;
    }
    return i - 1;
}
```

3.

a) There are a lot of errors in this code snippet, with the variable 'inside' undeclared, there must be parentheses before the 'function' variable, and that there are stray symbols after the variables 'fun' and 'bored' such as a forward slash and unpaired quotation mark. These errors may be corrected by possibly arranging some elements of the snippet correctly that there are no unmatched or stray symbols in the code. Since 'inside' and 'function' as variables seem to be unnecessary on the code, it can also be possibly omitted.

b) This snippet merely lacks the main function, which must be added for the program to correctly execute.

c) This snippet has an identifier before the braces to properly function, as well as 'a' redeclared in line 4. These errors may be corrected by maintaining a single declaration for 'a' as a float.

d) This snippet's errors lie at the order of the declaration of the variables as well as the lack of semicolon on line 2. This can easily be corrected by moving the declaration for variables 'a', 'b', and 'c' from line 3 to line 1 and making sure that semicolons are added for each line of the code.

4.

a) int array[5] = {1,2,3,4,5};
b) int *ptr;
c) ptr = array[0]
d) printf("Address of variable num is: %p", *ptr);
e) printf("Address of variable num is: %p", *array[0]);
f)
   1) int (*ptr+2) = array[2];
   2) int ptr+2(*array[2]);
   3) int *ptr+2(array[2]);
   4) int *ptr+2;

g) The address referenced by ptr+2 is array[2], with the value 2 being stored at that address as referenced at the aforementioned array.

***Unfortunately, I have no answers for Part 2 - Item #5 and Part III – Item #2. I apologize and take full responsibility of this shortcoming, Ma'am.***

**Francis Wilfred B. Olilang**
**CMSC 21 - Second Long Examination Answer Sheet**
**May 22, 2022**

## Part III – Item 1

```
//Part III - Item 1
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdbool.h>

void scan_word (int occurrences[26]);
bool is_anagram(int occurrences1[26], int
occurrences2[26]);

int i;

int main(void){
    int occurrences1[26] = {0},
    occurrences2[26] = {0};

    printf("Please enter the first word.\n> ");
    scan_word(occurrences1);

    printf("Please enter the second word.\n> ");
    scan_word(occurrences2);

    if(is_anagram(occurrences1, occurrences2)){
        printf("\nYay! The words are anagrams!\n");
        system("PAUSE");
        return 0;
    }
    else{
        printf("\nUh oh! It seems that the words are not
anagrams.\n");
        system("PAUSE");
        return 0;
    }
}
```

***Unfortunately, I have no answers for Part 2 - Item #5 and Part III – Item #2. I apologize and take full responsibility of this shortcoming, Ma'am.***

**Francis Wilfred B. Olilang**
**CMSC 21 - Second Long Examination Answer Sheet**
**May 22, 2022**

```
void scan_word(int occurrences[26]){
    char c;
    while ((c = getchar()) != '\n'){
        if(isalpha(c)){
            occurrences[toupper(c) - 'A']++;
        }
    }
}

bool is_anagram(int occurrences1[26], int
occurrences2[26]){
    for(i = 0; i < 26; i++){
        if(occurrences1[i] != occurrences2[i]){
            return false;
        }
    }
}
```

GitHub Code Link: https://github.com/front-git/CMSC21/blob/main/SecondLE/Olilang_SecondLE_TestIII_Item1.c

***Unfortunately, I have no answers for Part 2 - Item #5 and Part III – Item #2. I apologize and take full responsibility of this shortcoming, Ma'am.***