Github link for Test 3 Application codes: https://github.com/zaxepaz/CMSC-21/tree/main/Second%20Long%20Exam

I.

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

II.

1.

Arrays are arranged linearly by the compiler. If the first dimension is left unspecified or specified but the other dimensions are left unspecified, the compiler will not know how to divide and store the elements of an array into specific memory blocks for each layer in the first dimension of an array.

2. a. bool isPalindrome (char *string)
   b. float computeAverage (float arr[20])
   c. void reverseSentence(void)
   d. float squareRoot (int n)

3.
   a.) Since C does not allow nested functions, there is an error in the code and function int bored(void) will not work. For this code snippet to work, we can separate the two functions and just call bored function instead of declaring it inside fun function.

   Correct code would look like:

```
int bored(void){
    printf("%s", "Inside function bored\n");
    }

int fun(void){
    printf("%s", "Inside function fun\n");
    bored();
}
```

b.) The error in this code is that the function is declared as an integer, but it does not return any integer value. For this code to be correct, we can include a line return result;
The correct code would look like:

```
int product (int a, int b){
    int result = a * b;
    return result;

}
```

c.) The error in the code snippet is that there is ; after float(a) which causes the program to ignore the code found inside the function. The other error is that a is redeclared as a float.

To fix the code, ; should be removed and also float a;

Correct code will look like:

```
void fun (float a){

    printf("%f", a);

}
```

d.) The error in the code is that the function states it will return void but there is a return command located in the function that returns a value. To fix this code, we can remove return total;

There is also no ; after printf("%s", "Enter three integers: ")
So we can place ; after this line of code.

Correct code would look like:

```c
void sum(void){

    printf("%s", "Enter three integers: ");
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);
    int total = a + b + c;
    printf("Result is %d", total);
}
```

4.

a.) int array [SIZE] = {1,2,3,4,5};

b.) int *ptr

c.) ptr = array

d.) for (int i = 0; i < SIZE; i++){
printf("%d",*(ptr + i));
}

e.) for (int i = 0; i < SIZE; i++){

printf("%d",*(array + i));
}

f.) f.1) array[1]
f.2) *(array + 1)
f.3) ptr[1]
f.4) *(ptr + 1)

g.) ptr+2 would refer to 2508 if the address of the first element of the array is 2500 in memory. The value stored at address ptr + 2 is 3.

5. a. xp = x;
   b. num = *xp
   c. num = xp[1]
   d. num++;

III.

1.

```c
#include <stdio.h>
#include <ctype.h> /* toupper, isalpha */
#include <stdbool.h>


void scan_word(int occurences[26]){//scan_word function

    char c;//declares c as char type

    /*uses while loop to obtain every character from user input, then check if character is in alphabet.
    If character is in alphabet, it gets the index value of the given letter in the ocurrences array and increments it by 1.*/
    while ((c = getchar()) != '\n') {
        if (isalpha(c)){
            occurences[toupper(c) - 'A']++;
        }
    }
}


bool is_anagram(int occurences1[26], int occurences2[26]){//is_anagram to compare occurences1 and occurences2 array from scan_word function.

    int i,sum1 = 0,sum2 = 0;//declares i,sum1,sum2 as integer type variables.


    /*for loop to check every element in occurence1 and occurence2. If element at occurences1[i] is the same in occurences2[i],both elements are set to 0.*/
    for (i=0; i<26; i++){

        if (occurences1[i] == occurences2[i]){
            occurences1[i] = 0;
            occurences2[i] = 0;
        }
    }

    /*after the previous for loop, this for loop is for taking the sum of both occurences arrays to check if the sum of both arrays are 0 or not.
    If the sum is equal to zero, that means there were elements that were set to 0 in both arrays meaning that the user input for the two words are anagrams.
    Otherwise, they aren't*/
    for (i = 0; i < 26; i++){
        sum1 += occurences1[i];
        sum2 += occurences2[i];
    }


    /*returns 1 if sum of both arrays is 0 and returns 0 if sum of both arrays is greater than 0*/
    if (sum1 + sum2 == 0){
        return 1;
    }

    else if(sum1 + sum2 > 0){
        return 0;
    }
}



int main(void) {

    /*declares i, j, final_answer, occurences1, and occurences2 as integer type variables and arrays.*/
    int i,j,final_answer,occurences1[26] = {0},occurences2[26] = {0};


    char c;//declares c as char type variable.


    /*Asks user to input first word and uses scan_word function using occurences1 array.*/
    printf("Enter first word: ");
    scan_word(occurences1);
```

```
67
68      /*Asks user to input second word and uses scan_word function using occurences2 array.*/
69      printf("Enter second word: ");
70      scan_word(occurences2);
71
72      /*uses variable final_answer to receive value from is_anagram function.*/
73      final_answer = is_anagram(occurences1,occurences2);
74
75
76      /*if final_answer is True, then the program prints that it is an anagram, otherwise, it prints it is not an anagram.*/
77      if (final_answer){
78          printf("\nAnagram.");
79      }
80
81      else {
82          printf("\nNot an anagram.");
83      }
84
85      return 0;
86  }
87
```

2.

```
1   #include <stdio.h>
2   #include <ctype.h> /* toupper, isalpha */
3   #include <stdbool.h>
4
5
6   void scan_word(int occurences[26]){//scan_word function
7
8       char c;//declares c as char type
9
10      int *ptr;//integer pointer.
11
12      ptr = occurences;//assigns the first element of array occurences to pointer ptr.
13
14      /*uses while loop to obtain every character from user input, then check if character is in alphabet.
15      If character is in alphabet, it gets the index value of the given letter in the ocurrences array using ptr pointer and increments it by 1.*/
16      while ((c = getchar()) != '\n') {
17          if (isalpha(c)){
18                  ptr[toupper(c) - 'A']++;
19          }
20      }
21  }
22
23
24  bool is_anagram(int occurences1[26], int occurences2[26]){//is_anagram to compare occurences1 and occurences2 array from scan_word function.
25
26      int i,*ptr1,*ptr2,sum1 = 0,sum2 = 0;//declares i,sum1,sum2 as integer type variables and *ptr1 and *ptr2 as integer pointers.
27
28      ptr1 = occurences1;//assigns the first element of array occurences1 to pointer ptr1.
29      ptr2 = occurences2;//assigns the first element of array occurences2 to pointer ptr2.
30
31      /*for loop to check every element in occurence1 and occurence2 using ptr1 and ptr2 as pointers respectively. If element at ptr1[i] is the same in ptr2[
32      for (i=0; i<26; i++){
```

```
        if (ptr1[i] == ptr2[i]){
            ptr1[i] = 0;
            ptr2[i] = 0;
        }
    }

    /*after the previous for loop, this for loop is for taking the sum of both occurences arrays to check if the sum of both arrays are 0 or not using ptr1 a
    If the sum is equal to zero, that means there were elements that were set to 0 in both arrays meaning that the user input for the two words are anagrams.
    Otherwise, they aren't*/
    for (i = 0; i < 26; i++){
        sum1 += ptr1[i];
        sum2 += ptr2[i];
    }


    /*returns 1 if sum of both arrays is 0 and returns 0 if sum of both arrays is greater than 0*/
    if (sum1 + sum2 == 0){
        return 1;
    }

    else if(sum1 + sum2 > 0){
        return 0;
    }
}
```

```
int main(void) {

    /*declares i, j, final_answer, occurences1, and occurences2 as integer type variables and arrays.*/
    int i,j,final_answer,occurences1[26] = {0},occurences2[26] = {0};


    char c;//declares c as char type variable.


    /*Asks user to input first word and uses scan_word function using occurences1 array.*/
    printf("Enter first word: ");
    scan_word(occurences1);

    /*Asks user to input second word and uses scan_word function using occurences2 array.*/
    printf("Enter second word: ");
    scan_word(occurences2);

    /*uses variable final_answer to receive value from is_anagram function.*/
    final_answer = is_anagram(occurences1,occurences2);


    /*if final_answer is True, then the program prints that it is an anagram, otherwise, it prints it is not an anagram.*/
    if (final_answer){
        printf("\nAnagram.");
    }

    else {
        printf("\nNot an anagram.");
    }

    return 0;
}
```