RIAN LUIS C. MABAIT BS in Computer Science – I

ASSIGNMENT 11

1. No pointers

CODE SNIPPET

```c
#include <stdio.h>
#include <stdbool.h>
#include <ctype.h>


bool is_anagram(int occurrences1[26], int occurrences2[26]){
    // for loop that checks each letter in the words
    for (int i = 0; i < 26; i++){
        // compares the counts & checks if there are differences in the occurrence of letters
        if (occurrences1[i] != occurrences2[i]){
            return false; // returns false if not an anagram
        }
    }
    // returns true if it's an anagram
    return true;
}

void scan_word(int occurrences[26]){
    char c; // variable declaration
    // scans the user input
    for (c = getchar(); c != '\n'; c = getchar()){
        // checks if it is a character or not
        if (isalpha(c)){
            // shifts the index of occurences to be a number between [0-25]
            // by subtracting the ASCII value of 'A' which is 65
            // from the current value of that character
            // and takes the index from [0-25] and increment it
            occurrences[toupper(c) - 'A']++; // uses toupper to not make it case sensitive
        }
    }
}

int main(){

int i = 0, occurrences1[26] = {0}, occurrences2[26] = {0}; // counter arrays that initializes at 0
    bool same; // variable declaration at type bool

    while (i < 2){

        // prompts the user to enter the first and second words
        // and function calls correspondingly to scan occurences1 & 2
        printf("Enter first word: ");
        scan_word(occurrences1);

        printf("Enter second word: ");
        scan_word(occurrences2);

        same = is_anagram(occurrences1,occurrences2); // stores the function

        // function call at variable same
        if (same){
            printf("The words are anagrams.\n");
        }
        else{
            printf("The words are not anagrams.\n");
        }

    printf("\n");
    i++;
    }

    return 0;

}
```

SAMPLE OUTPUTS

Anagram:

```
PS C:\Users\RIAN\Desktop\CMSC21\Lecture11\Assignments>
Enter first word: Salient
Enter second word: tenisla
The words are anagrams.

Enter first word: smartest
Enter second word: mattress
The words are anagrams.
```

Not Anagrams:

```
PS C:\Users\RIAN\Desktop\CMSC21>
Enter first word: Dumbest
Enter second word: stumble
The words are not anagrams.

Enter first word: salient
Enter second word: silent
The words are not anagrams.
```

Anagram & Not Anagram:

```
PS C:\Users\RIAN\Desktop\CMSC21\L
Enter first word: sILENT
Enter second word: lISTEN
The words are anagrams.

Enter first word: anagram
Enter second word: grahams
The words are not anagrams.
```

2. Use of pointers

```c
#include <stdio.h>
#include <stdbool.h>
#include <ctype.h>

//function that checks if two arrays are anagrams
bool is_anagram(int* occurrences1, int* occurrences2){
    // for loop that checks each letter in the words
    for (int i = 0; i < 26; i++){

// compares the values at memory locations pointed by occurrences1 + i & occu
rrences2 + i
        if (*(occurrences1 + i) != *(occurrences2 + i)){
            return false; // returns false if not an anagram
        }
    }
    // returns true if it's an anagram
    return true;
}

// function that scans input from the user and updates the array
void scan_word(int* occurrences){
    char c; // variable declaration
    // scans the user input
    for (c = getchar(); c != '\n'; c = getchar()){
        // checks if it is a character or not
        if (isalpha(c)){

            (*(occurrences + (toupper(c) - 'A')))+=1;
 // uses toupper to not make it case sensitive and subtract 'A' to get index
        }
    }
}

int main(){

int i = 0, occurrences1[26] = {0}, occurrences2[26] = {0};
 // counter arrays that initializes at 0
    bool same; // variable declaration at type bool

    while (i < 2){

        // prompts the user to enter the first and second words
        // and function calls correspondingly to scan occurences1 & 2
        printf("Enter first word: ");
        scan_word(occurrences1);

        printf("Enter second word: ");
        scan_word(occurrences2);

        same = is_anagram(occurrences1,occurrences2); // stores the function

        // function call at variable same
        if (same){
            printf("The words are anagrams.\n");
        }
        else{
            printf("The words are not anagrams.\n");
        }

    printf("\n");
    i++;
    }
    return 0;

}
```

SAMPLE OUTPUTS

Anagram:

```
PS C:\Users\RIAN\Desktop\CMSC21\
Enter first word: smARTEST
Enter second word: MATTRESS
The words are anagrams.

Enter first word: race
Enter second word: care
The words are anagrams.
```

Not Anagram:

```
PS C:\Users\RIAN\Desktop\CMSC21\L
Enter first word: race
Enter second word: HIDE
The words are not anagrams.

Enter first word: league
Enter second word: fish
The words are not anagrams.
```

Anagram & Not Anagram:

```
PS C:\Users\RIAN\Desktop\CMSC21\
Enter first word: hEART
Enter second word: EArth
The words are anagrams.

Enter first word: scare
Enter second word: pleasE
The words are not anagrams.
```