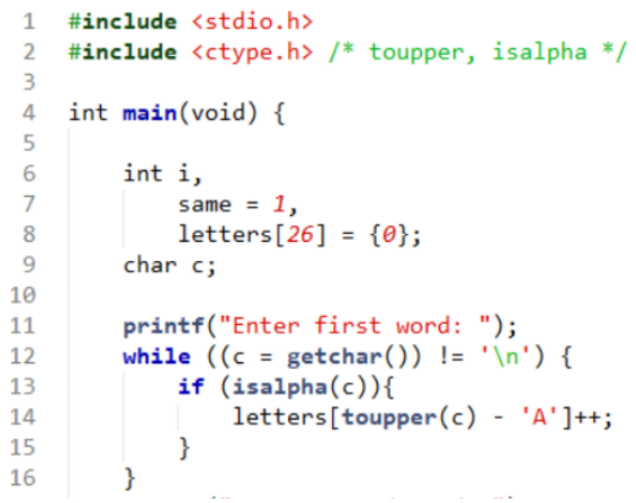
CMSC 21

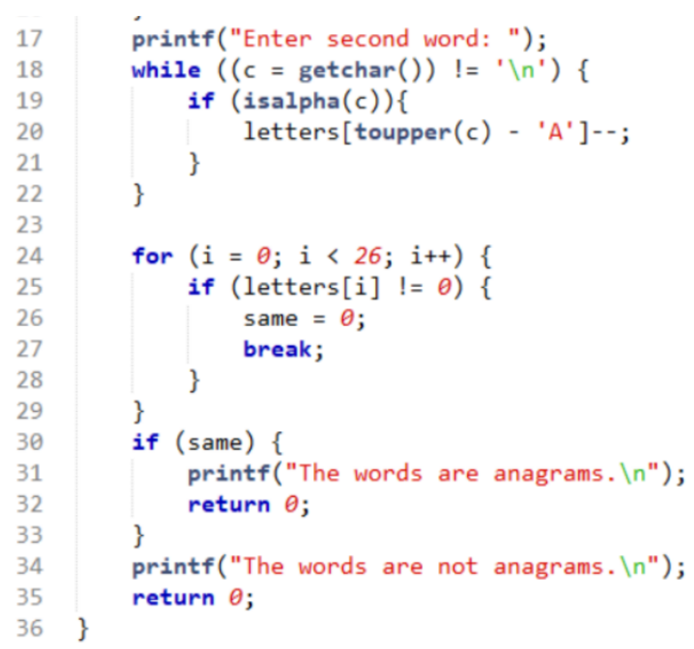
2nd Semester AY 2022-2023

LECTURE 11

Jhoanna R. Olana - 202211140



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

int main(void) {

    int i,
        same = 1,
        letters[26] = {0};
    char c;

    printf("Enter first word: ");
    while ((c = getchar()) != '\n') {
        if (isalpha(c)){
            letters[toupper(c) - 'A']++;
        }
    }
```

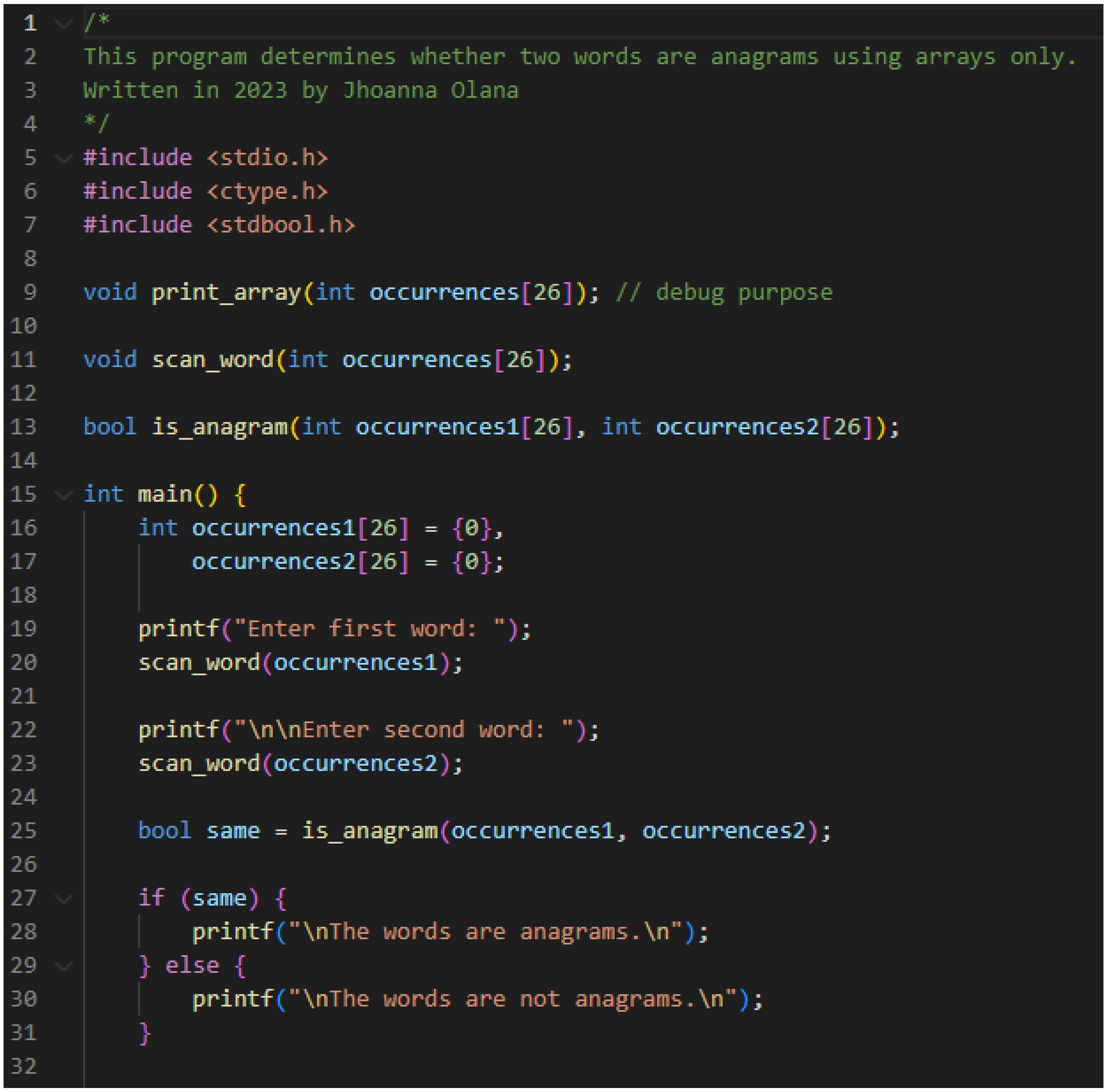


```c
    printf("Enter second word: ");
    while ((c = getchar()) != '\n') {
        if (isalpha(c)){
            letters[toupper(c) - 'A']--;
        }
    }

    for (i = 0; i < 26; i++) {
        if (letters[i] != 0) {
            same = 0;
            break;
        }
    }
    if (same) {
        printf("The words are anagrams.\n");
        return 0;
    }
    printf("The words are not anagrams.\n");
    return 0;
}
```

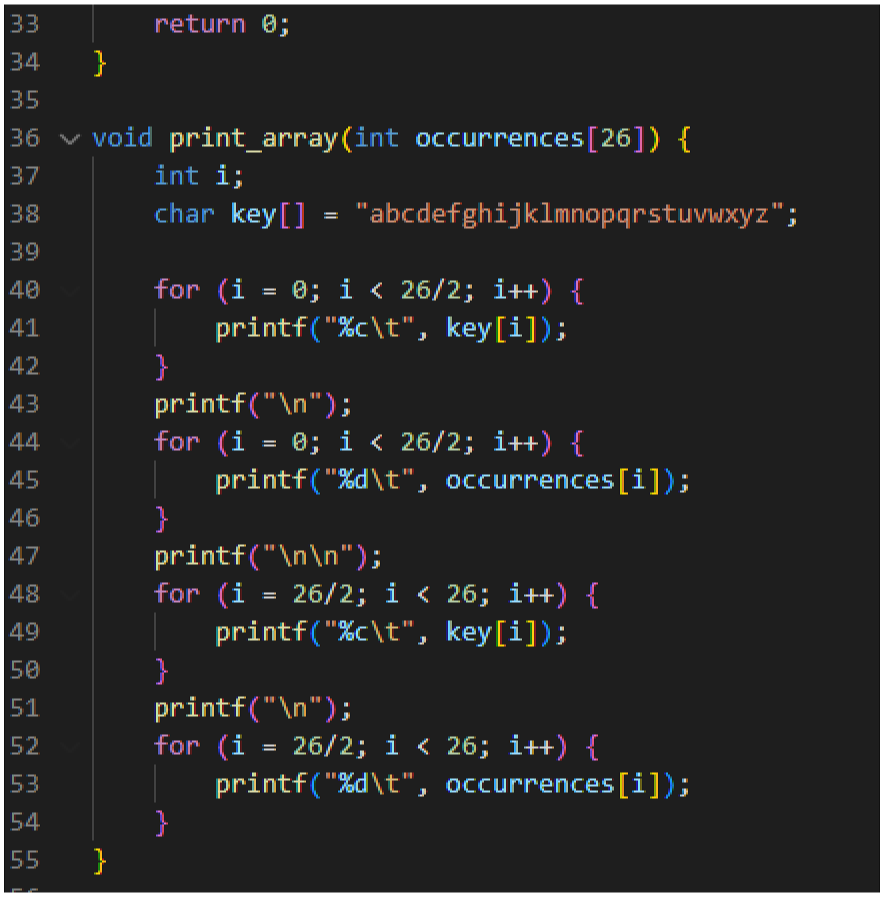
Item 1:

Your task:

- Modify the anagram code above such that following functions are added:
  - void scan_word(int occurrences[26]);
  - bool is_anagram(int occurrences1[26], int occurrences2[26]);

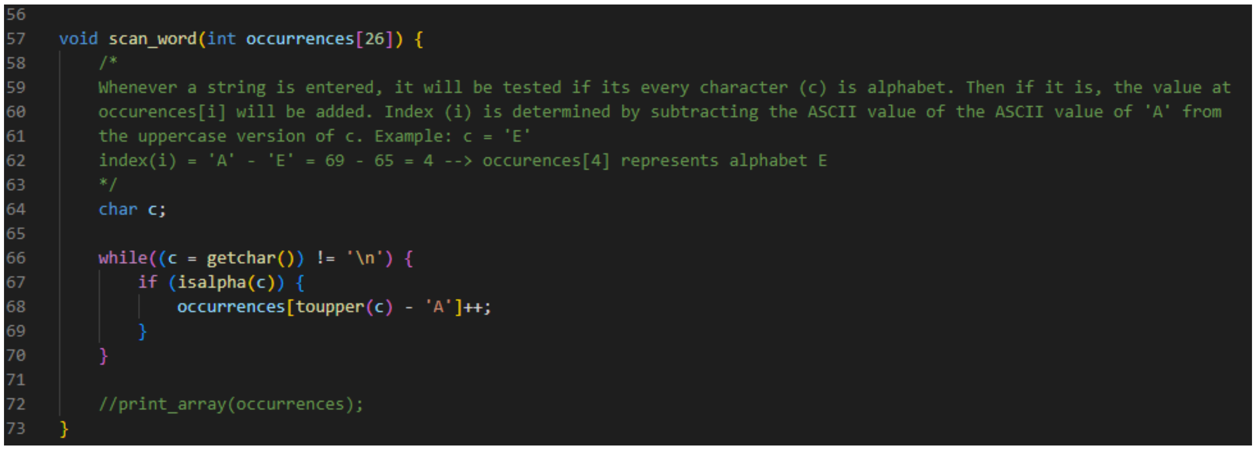


```c
/*
This program determines whether two words are anagrams using arrays only.
Written in 2023 by Jhoanna Olana
*/
#include <stdio.h>
#include <ctype.h>
#include <stdbool.h>

void print_array(int occurrences[26]); // debug purpose

void scan_word(int occurrences[26]);

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

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

    printf("Enter first word: ");
    scan_word(occurrences1);

    printf("\n\nEnter second word: ");
    scan_word(occurrences2);

    bool same = is_anagram(occurrences1, occurrences2);

    if (same) {
        printf("\nThe words are anagrams.\n");
    } else {
        printf("\nThe words are not anagrams.\n");
    }

```

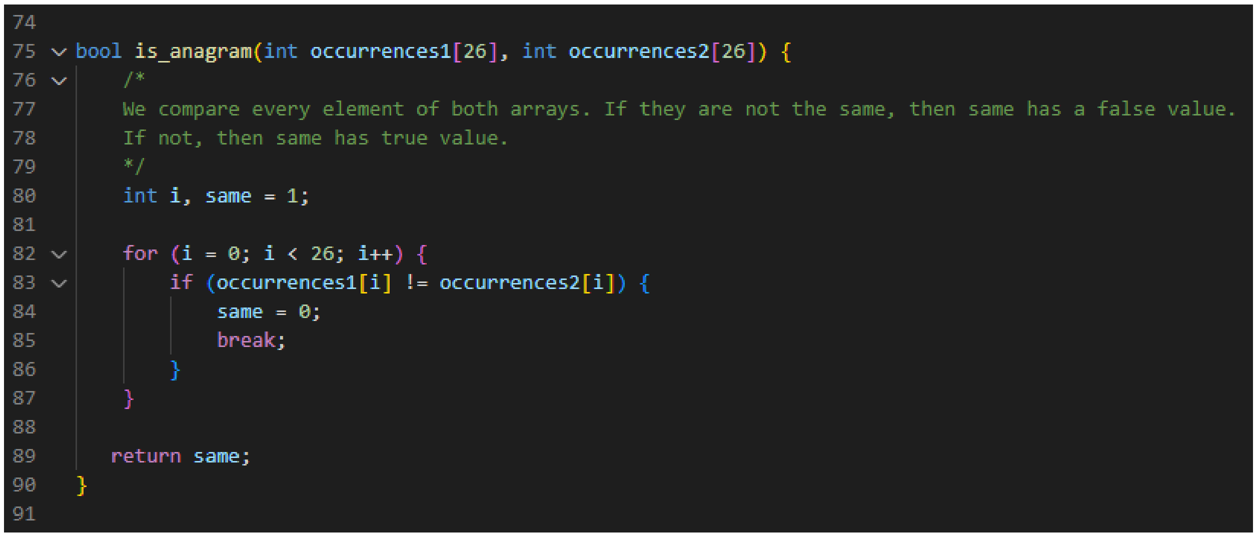


```c
    return 0;
}

void print_array(int occurrences[26]) {
    int i;
    char key[] = "abcdefghijklmnopqrstuvwxyz";

    for (i = 0; i < 26/2; i++) {
        printf("%c\t", key[i]);
    }
    printf("\n");
    for (i = 0; i < 26/2; i++) {
        printf("%d\t", occurrences[i]);
    }
    printf("\n\n");
    for (i = 26/2; i < 26; i++) {
        printf("%c\t", key[i]);
    }
    printf("\n");
    for (i = 26/2; i < 26; i++) {
        printf("%d\t", occurrences[i]);
    }
}
```

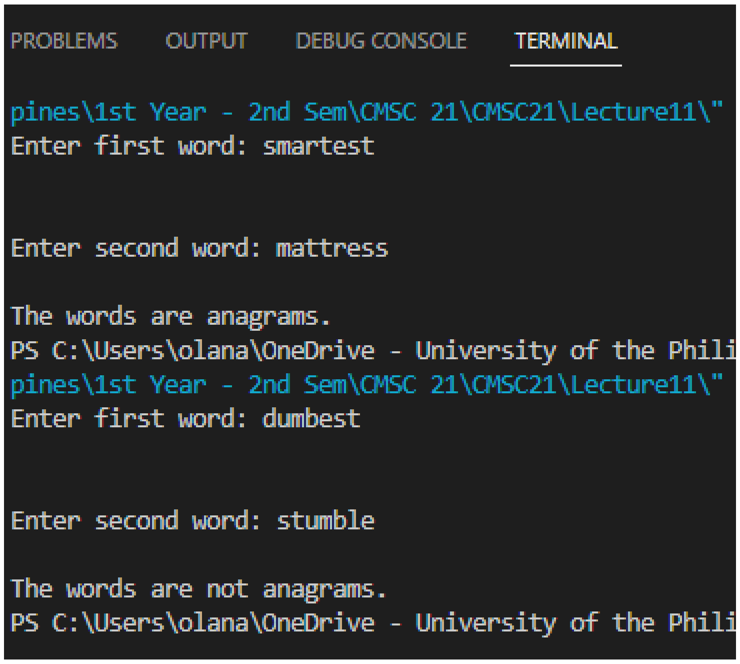
// for debugging purposes



```c

void scan_word(int occurrences[26]) {
    /*
    Whenever a string is entered, it will be tested if its every character (c) is alphabet. Then if it is, the value at
    occurences[i] will be added. Index (i) is determined by subtracting the ASCII value of the ASCII value of 'A' from
    the uppercase version of c. Example: c = 'E'
    index(i) = 'A' - 'E' = 69 - 65 = 4 --> occurences[4] represents alphabet E
    */
    char c;

    while((c = getchar()) != '\n') {
        if (isalpha(c)) {
            occurrences[toupper(c) - 'A']++;
        }
    }

    //print_array(occurrences);
}
```



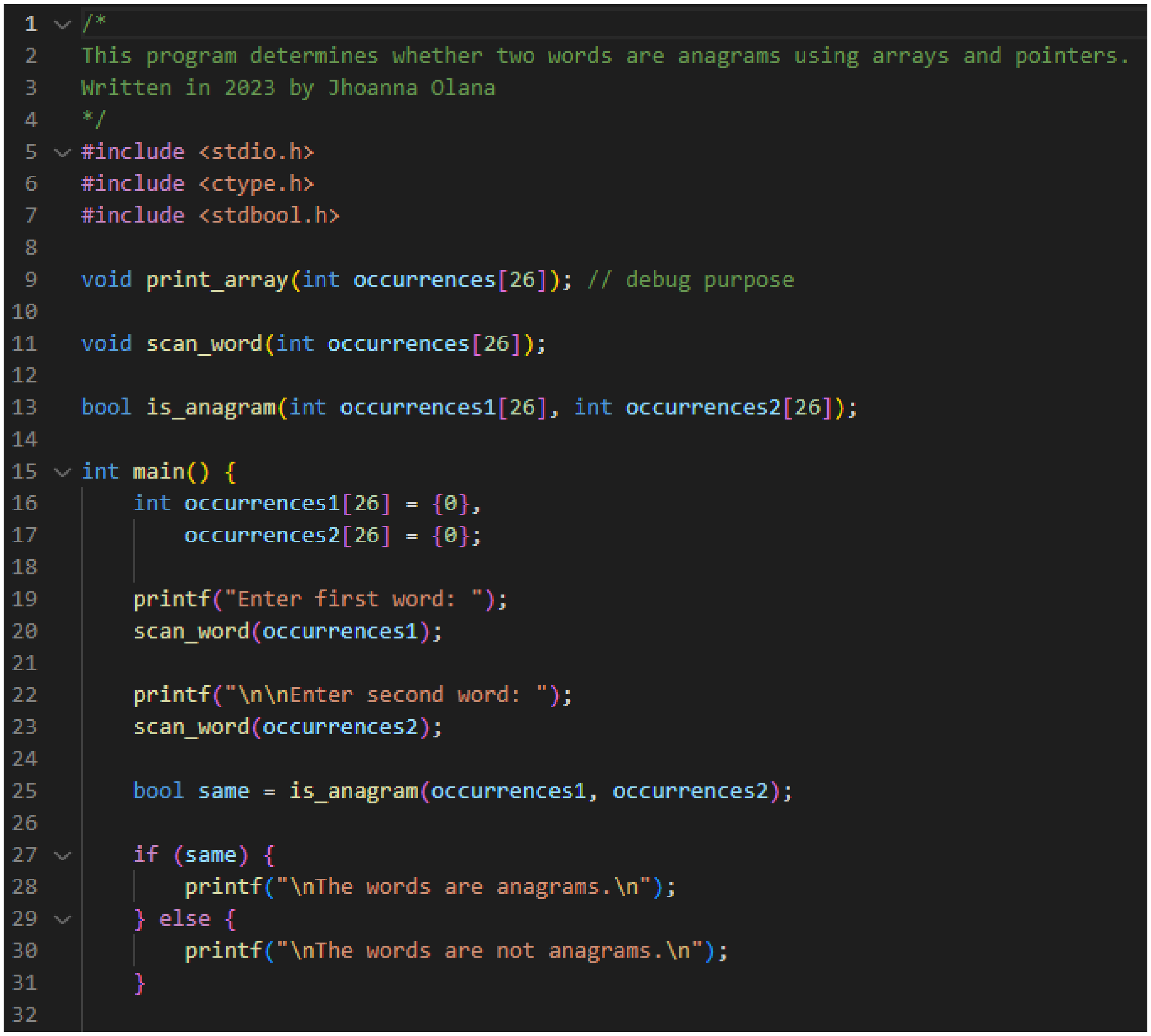
```c

bool is_anagram(int occurrences1[26], int occurrences2[26]) {
    /*
    We compare every element of both arrays. If they are not the same, then same has a false value.
    If not, then same has true value.
    */
    int i, same = 1;

    for (i = 0; i < 26; i++) {
        if (occurrences1[i] != occurrences2[i]) {
            same = 0;
            break;
        }
    }

   return same;
}

```

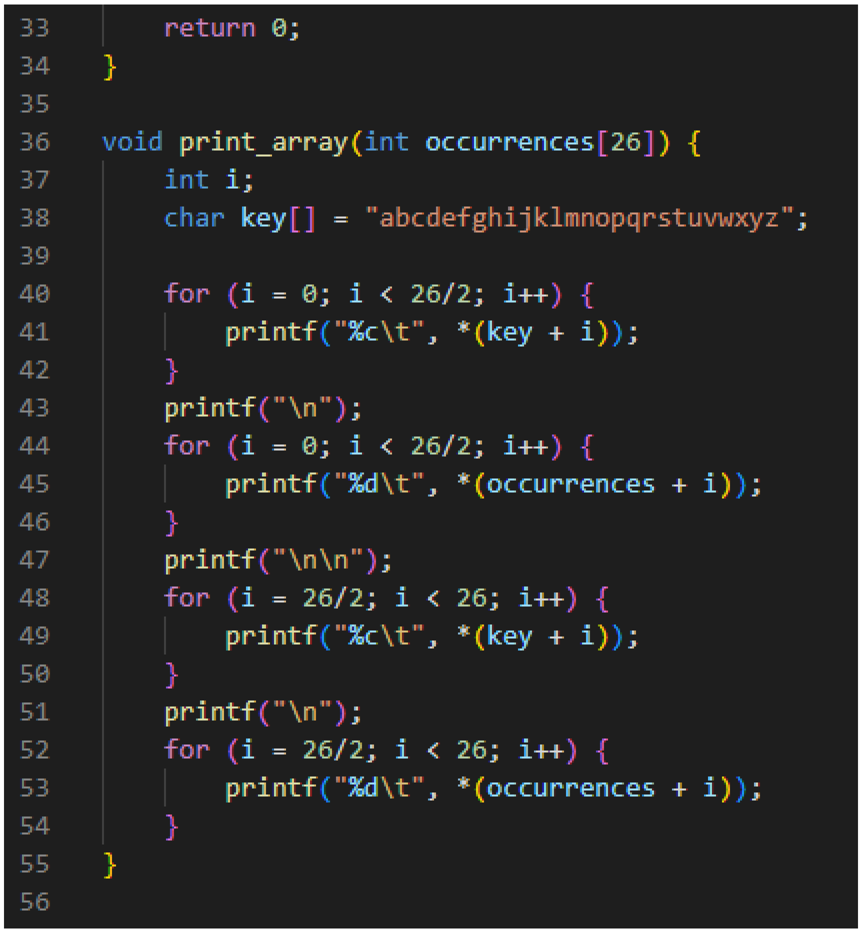


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

pines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture11\"
Enter first word: smartest


Enter second word: mattress

The words are anagrams.
PS C:\Users\olana\OneDrive - University of the Phili
pines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture11\"
Enter first word: dumbest


Enter second word: stumble

The words are not anagrams.
PS C:\Users\olana\OneDrive - University of the Phili
```

Item 2:

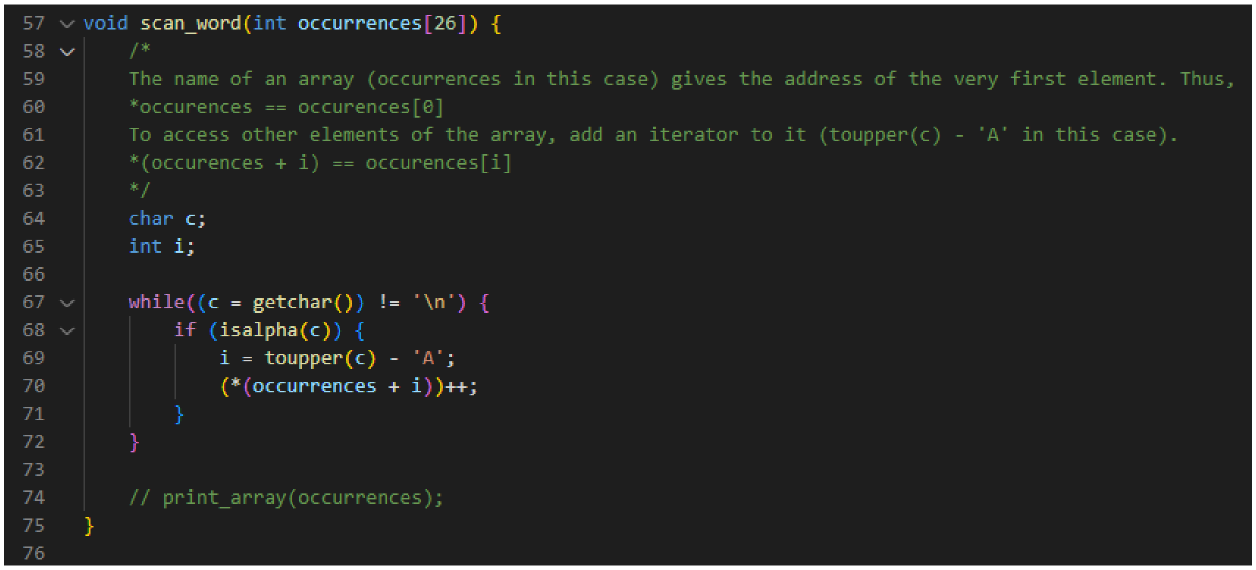
Convert your source code in Application Item #1 such that you operate on the arrays using pointers.


```c
/*
This program determines whether two words are anagrams using arrays and pointers.
Written in 2023 by Jhoanna Olana
*/
#include <stdio.h>
#include <ctype.h>
#include <stdbool.h>

void print_array(int occurrences[26]); // debug purpose

void scan_word(int occurrences[26]);

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

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

    printf("Enter first word: ");
    scan_word(occurrences1);

    printf("\n\nEnter second word: ");
    scan_word(occurrences2);

    bool same = is_anagram(occurrences1, occurrences2);

    if (same) {
        printf("\nThe words are anagrams.\n");
    } else {
        printf("\nThe words are not anagrams.\n");
    }

```

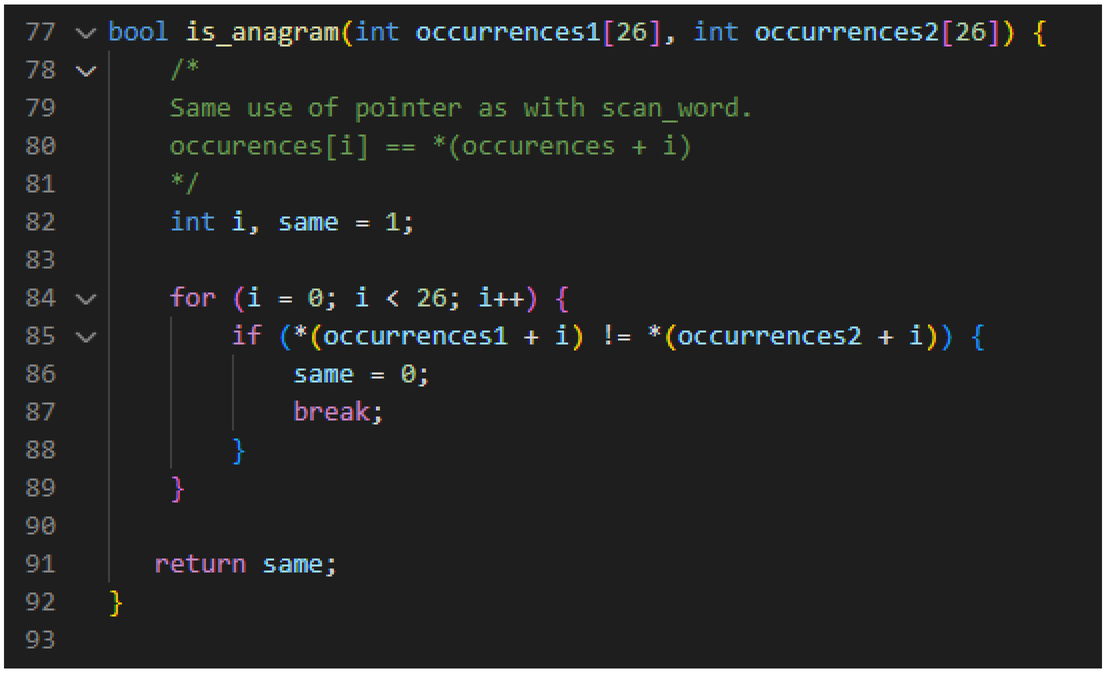


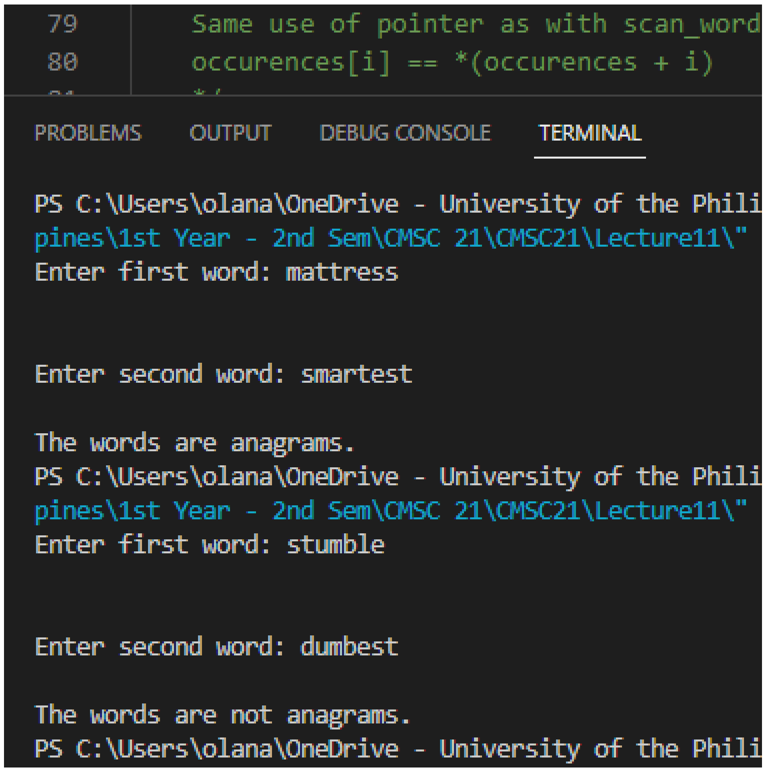
```c
    return 0;
}

void print_array(int occurrences[26]) {
    int i;
    char key[] = "abcdefghijklmnopqrstuvwxyz";

    for (i = 0; i < 26/2; i++) {
        printf("%c\t", *(key + i));
    }
    printf("\n");
    for (i = 0; i < 26/2; i++) {
        printf("%d\t", *(occurrences + i));
    }
    printf("\n\n");
    for (i = 26/2; i < 26; i++) {
        printf("%c\t", *(key + i));
    }
    printf("\n");
    for (i = 26/2; i < 26; i++) {
        printf("%d\t", *(occurrences + i));
    }
}

```

// for debugging purposes



```c
void scan_word(int occurrences[26]) {
    /*
    The name of an array (occurrences in this case) gives the address of the very first element. Thus,
    *occurences == occurences[0]
    To access other elements of the array, add an iterator to it (toupper(c) - 'A' in this case).
    *(occurences + i) == occurences[i]
    */
    char c;
    int i;

    while((c = getchar()) != '\n') {
        if (isalpha(c)) {
            i = toupper(c) - 'A';
            (*(occurrences + i))++;
        }
    }

    // print_array(occurrences);
}

```


```c
bool is_anagram(int occurrences1[26], int occurrences2[26]) {
    /*
    Same use of pointer as with scan_word.
    occurences[i] == *(occurences + i)
    */
    int i, same = 1;

    for (i = 0; i < 26; i++) {
        if (*(occurrences1 + i) != *(occurrences2 + i)) {
            same = 0;
            break;
        }
    }

   return same;
}

```


```
79      Same use of pointer as with scan_word
80      occurences[i] == *(occurences + i)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
PS C:\Users\olana\OneDrive - University of the Phili
pines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture11\"
Enter first word: mattress


Enter second word: smartest

The words are anagrams.
PS C:\Users\olana\OneDrive - University of the Phili
pines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture11\"
Enter first word: stumble


Enter second word: dumbest

The words are not anagrams.
PS C:\Users\olana\OneDrive - University of the Phili
```

Github Link: CMSC21/Lecture11 at main · jrolana/CMSC21 · GitHub