S.No: 1	Exp. Name: ***Project Module***	Date: 2024-06-14

Aim:

Project Module

Source Code:

hello.c

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_WORDS 1000
#define MAX_WORD_LENGTH 500

// Function to remove leading and trailing whitespace from a string
void trim(char *str) {
        int start = 0, end = strlen(str) - 1;
        while (isspace((unsigned char)str[start])) {
                start++;
        }
        while ((end >= start) && isspace((unsigned char)str[end])) {
                end--;
        }
        memmove(str, str + start, end - start + 1);
        str[end - start + 1] = '\0';
}

// Function to convert a string to lowercase
void toLowercase(char *str) {
        for (int i = 0; str[i]; i++) {
                str[i] = tolower((unsigned char)str[i]);
        }
}

int main() {
        char input[MAX_WORD_LENGTH];
        char words[MAX_WORDS][MAX_WORD_LENGTH];
        int frequency[MAX_WORDS] = {0};
        int wordCount = 0, wordStart = 0;
        printf("Enter a string: ");
        fgets(input, sizeof(input), stdin);
        input[strcspn(input, "\n")] = '\0'; // Remove newline character
        toLowercase(input);
        // Convert to lowercase
        // Parse the input string to extract words
        for (int i = 0; input[i]; i++) {
                if (!isalpha(input[i])) {
                        // Check if the character is not an alphabet
                        if (i > 0 && isalpha(input[i - 1])) {
                                // Check if the previous character was an alphabet
                                strncpy(words[wordCount], input + wordStart, i -
wordStart);
                                words[wordCount][i - wordStart] = '\0';
                                trim(words[wordCount]);
                                // Remove leading/trailing whitespace
                                wordCount++;
                        }
                        wordStart = i + 1;
                        // Move to the next word
                }
        }
        // Handle the last word if any
```

```
                strncpy(words[wordCount], input + wordStart, strlen(input) -
wordStart);
                words[wordCount][strlen(input) - wordStart] = '\0';
                trim(words[wordCount]);
                // Remove leading/trailing whitespace
                wordCount++;
        }
        // Count the frequency of each word
        for (int i = 0; i < wordCount; i++) {
                if (words[i][0] != '\0') {
                        // Skip empty words
                        frequency[i]++;
                        for (int j = i + 1; j < wordCount; j++)
                        {
                                if (strcmp(words[i], words[j]) == 0)
                                {
                                        frequency[i]++;
                                        words[j][0] = '\0';
                                        // Mark duplicate words as empty
                                }
                        }
                }
        }
        // Print the word frequencies
        printf("\nWord frequencies:\n");
        for (int i = 0; i < wordCount; i++) {
                if (words[i][0] != '\0') {
                        // Print only non-empty words
                        printf("%s: %d\n", words[i], frequency[i]);
                }
        }
        return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Hello World