**Assignment 11**

// String Operations on 'Manish Mahale' in C

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()

{

    char string[] = "Manish Mahale";

    char temp[100];

    char ch;

    int i, count = 0;

    printf("%s \n", string);

    printf("\n----------------------------------------------------------\n");

    // 1. Write a program to scan string from user then scan a single character and search it in a accepted string.

    printf("Enter a character to search: ");

    scanf(" %c", &ch);

    int found = 0;

    for (i = 0; string[i] != '\0'; i++)

    {

        if (string[i] == ch)

        {

            found = 1;

            break;

        }

    }

    if (found)

    {

        printf("'%c' found in the string.\n", ch);

    }

    else

    {

        printf("'%c' not found in the string.\n", ch);

    }

    printf("\n----------------------------------------------------------\n");

    // 2. WAP Replace all Occurrences of ‘a’ with $ in a String

    strcpy(temp, string);

    for (i = 0; temp[i] != '\0'; i++)

    {

        if (temp[i] == 'a')

        {

            temp[i] = '$';

        }

    }

    printf("Updated String (a->$): %s\n", temp);

    printf("\n----------------------------------------------------------\n");

    // 3. WAP to Remove the nth Index Character from a Non-Empty String

    int n;

    printf("Enter the index to remove: ");

    scanf("%d ", &n);

    strcpy(temp, string);

    int len = strlen(temp);

    if (n >= 0 && n < len)

    {

        for (i = n; i < len; i++)

        {

            temp[i] = temp[i + 1];

        }

        printf("String after removal: %s\n", temp);

    }

    else

    {

        printf("Invalid index.\n");

    }

    printf("\n----------------------------------------------------------\n");

    // 4. WAP to Form a New String where the First Character and the Last Character have been Exchanged

    strcpy(temp, string);

    len = strlen(temp);

    if (len >= 2)

    {

        char t = temp[0];

        temp[0] = temp[len - 1];

        temp[len - 1] = t;

    }

    printf("Swapped String: %s\n", temp);

    printf("\n----------------------------------------------------------\n");

    // 5. WAP to Count the Number of Vowels in a String

    count = 0;

    for (i = 0; string[i] != '\0'; i++)

    {

        char c = tolower(string[i]);

        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')

        {

            count++;

        }

    }

    printf("Number of vowels: %d\n", count);

    printf("\n----------------------------------------------------------\n");

    // 6. WAP to Take in a String and Replace Every Blank Space with special symbol.

    strcpy(temp, string);

    for (i = 0; temp[i] != '\0'; i++)

    {

        if (temp[i] == ' ')

        {

            temp[i] = '#';

        }

    }

    printf("Updated String (space->#): %s\n", temp);

    printf("\n----------------------------------------------------------\n");

    // 7. WAP to Remove the Characters of Odd Index Values in a String

    int j = 0;

    for (i = 0; string[i] != '\0'; i++)

    {

        if (i % 2 == 0)

        {

            temp[j++] = string[i];

        }

    }

    temp[j] = '\0';

    printf("String with odd index characters removed: %s\n", temp);

    printf("\n----------------------------------------------------------\n");

    // 8. WAP to Calculate the Number of Words Present in a String

    count = 1;

    for (i = 0; string[i] != '\0'; i++)

    {

        if (string[i] == ' ')

            count++;

    }

    printf("Number of words: %d\n", count);

    printf("\n----------------------------------------------------------\n");

    // 9. WAP to Take in Two Strings and Display the Larger String without Using Built-in Functions

    char string1[] = "Manish";

    char string2[] = "Mahale";

    int len1 = 0, len2 = 0;

    for (i = 0; string1[i] != '\0'; i++)

    {

        len1++;

    }

    for (i = 0; string2[i] != '\0'; i++)

    {

        len2++;

    }

    if (len1 > len2)

    {

        printf("Larger String: %s\n", string1);

    }

    else if (len2 > len1)

    {

        printf("Larger String: %s\n", string2);

    }

    else

    {

        printf("Both strings are equal in length.\n");

    }

    printf("\n----------------------------------------------------------\n");

    // 10. Write a program to check the string is palindrome or not.

    char check[100];

    printf("Enter a string to check palindrome: ");

    scanf("%s", check);

    len = strlen(check);

    int is\_palindrome = 1;

    for (i = 0; i < len / 2; i++)

    {

        if (check[i] != check[len - 1 - i])

        {

            is\_palindrome = 0;

            break;

        }

    }

    if (is\_palindrome)

    {

        printf("Palindrome\n");

    }

    else

    {

        printf("Not palindrome\n");

    }

    printf("\n----------------------------------------------------------\n");

    return 0;

}