ADVANCED C PROGRAMMING SLIPS SOLUTION

AM

ADVANCED C PROGRAMMING UNIVERSITY SLIP SOLUTTIONS

----------------------------------SLIP 02---------------------------

Q.1 Write C program to find maximum of two numbers using macros.

```
#include <stdio.h>

// Macro to find maximum of two numbers
#define MAX(a, b) ((a) > (b) ? (a) : (b))

int main() {
    int num1, num2, maximum;

    // Input two numbers
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    // Use macro to find maximum
    maximum = MAX(num1, num2);

    // Output the result
    printf("The maximum of %d and %d is %d\n", num1, num2, maximum);

    return 0;
}
```

Q.2 Write a function which takes a string as parameter and returns the same string in upper case (use pointer). Accept and Display the converted string in main only.

```
#include <stdio.h>

// Function to convert a string to uppercase using pointers
char* toUpperCase(char *str) {
    char *ptr = str;
    while (*ptr != '\0') {
        // If character is lowercase, convert to uppercase
        if (*ptr >= 'a' && *ptr <= 'z') {
            *ptr = *ptr - 32;
        }
        ptr++;
    }
    return str;
}

int main() {
    char str[100];

    // Accept a string from user
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);  // safer input method

    // Convert to uppercase
    toUpperCase(str);

    // Display the converted string
    printf("Uppercase string: %s", str);

    return 0;
}
```

---------------SLIP 03--

Q.1 Write a c program to accept strings and store it in an array. Display array elements, in reverse order.

```
#include <stdio.h>

int main() {
    char str[5][100];  // Array to store 5 strings (each up to 99
characters)
    int i;

    // Accept 5 strings using scanf
    printf("Enter 5 strings (one word each):\n");
    for (i = 0; i < 5; i++) {
        printf("String %d: ", i + 1);
        scanf("%s", str[i]);  // Reads one word (no spaces)
    }

    // Display strings in reverse order
    printf("\nStrings in reverse order:\n");
    for (i = 4; i >= 0; i--) {
        printf("%s\n", str[i]);
    }

    return 0;
}
```

```
/*------------output--------------------------
[?2004l
Enter 5 strings (one word each):
String 1: sachin
String 2: reena
String 3: isha
String 4: manohar
String 5: mohit

Strings in reverse order:
mohit
```

```
manohar
isha
reena
sachin


*/
```

Q.2 Write a function to compute area and perimeter of a circle. Accept required input and display desired output in main function.(Hint: use call by reference)

```c
#include <stdio.h>
#define PI 3.14

// Function to compute area and perimeter using pointers
void computeCircle(float radius, float *area, float *perimeter) {
    *area = PI * radius * radius;
    *perimeter = 2 * PI * radius;
}

int main() {
    float radius, area, perimeter;

    // Input radius
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    // Call function (call by reference)
    computeCircle(radius, &area, &perimeter);

    // Display results
    printf("Area = %.2f\n", area);
    printf("Perimeter = %.2f\n", perimeter);

    return 0;
}
```

---------------SLIP 05--

Q.1 Write C program to check given number is even or odd using macros.

```
#include <stdio.h>

// Macro to check even or odd
#define IS_EVEN(n) ((n) % 2 == 0 ? "Even" : "Odd")

int main() {
    int num;

    // Input number
    printf("Enter a number: ");
    scanf("%d", &num);

    // Use macro to check even or odd
    printf("The number %d is %s.\n", num, IS_EVEN(num));

    return 0;
}
```

Q.2 Write a program which accepts a string and two characters as command line arguments and replace all occurrences of the first character by the second.

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

int main(int argc, char *argv[]) {
    // Check if enough arguments are passed
    if (argc != 4) {
        printf("Usage: %s <string> <char_to_replace>
<replacement_char>\n", argv[0]);
        return 1;
    }

    char *str = argv[1];
    char ch1 = argv[2][0]; // character to be replaced
    char ch2 = argv[3][0]; // replacement character
    // Replace all occurrences of ch1 with ch2
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == ch1) {
            str[i] = ch2;
        }
    }

    // Display the modified string
    printf("Modified string: %s\n", str);

    return 0;
}
```

Note : If u r using Linux Terminal for execution of command line argument then
First compile your program then for execution type following command

```
./a.out  Rajmata  a b
```

It give output

```
Rbjmbtb
```

And if u are using online gdb compiler for then
Enter Rajmata a b in
Command line argument text box and press button run.

-----------------------------------SLIP 08----------------------------

Q. 1Write a C Program to check whether a number is positive, negative or zero using macros.

```
#include <stdio.h>

// Macro to check the number type
#define CHECK_NUMBER(n) ((n) > 0 ? "Positive" : ((n) < 0 ? "Negative"
: "Zero"))

int main() {
    int num;

    // Input number from user
    printf("Enter a number: ");
    scanf("%d", &num);

    // Use macro to check and display result
    printf("The number %d is %s.\n", num, CHECK_NUMBER(num));

    return 0;
}
```

Q.2Write a program to accept a string and character to be searched. The program will call a function to search the occurrence of the character in the string and return its position. Function should return -1 if the character is not found in the string. Display the position of character if found in main () function.

```
#include <stdio.h>

// Function to search for a character in a string
int searchChar(char str[], char ch) {
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == ch) {
            return i;  // Return index if found
        }
    }
    return -1;  // Return -1 if not found
}

int main() {
    char str[100], ch;
    int position;

    // Accept string (one word only) using scanf
    printf("Enter a string (no spaces): ");
    scanf("%s", str);

    // Accept character to search
    printf("Enter a character to search: ");
    scanf(" %c", &ch);  // space before %c skips leftover newline

    // Call function to search character
    position = searchChar(str, ch);

    // Display result
    if (position == -1) {
        printf("Character '%c' not found in the string.\n", ch);
    } else {
        printf("Character '%c' found at position %d (0-based
index).\n", ch, position);
    }

    return 0;
}
```

----------------------SLIP 09 ------------------------------------

Q.1 Write a C program to display all command line argument passed to main in the reverse order.

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    // Check if arguments are passed
    if (argc < 2) {
        printf("No command line arguments passed.\n");
        return 0;
    }

    printf("Command line arguments in reverse order:\n");

    // Display arguments in reverse (excluding program name at 
argv[0])
    for (int i = argc - 1; i > 0; i--) {
        printf("%s\n", argv[i]);
    }

    return 0;
}
```

Note for execution
Type
./a.out 1 2 3 4

Q. 2 Write a program to compare two strings. If they are not equal display their length and if equal concatenate them.

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

int main() {
    char str1[100], str2[100], result[200];

    // Accept two strings from user
    printf("Enter first string: ");
    scanf("%s", str1);

    printf("Enter second string: ");
    scanf("%s", str2);

    // Compare the strings
    if (strcmp(str1, str2) == 0) {
        // If equal, concatenate them
        strcpy(result, str1);        // Copy first string to result
        strcat(result, str2);        // Concatenate second string
        printf("Strings are equal. Concatenated string: %s\n",
result);
    } else {
        // If not equal, display their lengths
        printf("Strings are not equal.\n");
        printf("Length of first string: %lu\n", strlen(str1));
        printf("Length of second string: %lu\n", strlen(str2));
    }

    return 0;
}
```

---SLIP 11----------------------

Q. 1 Write a program to find maximum of two integers using pointer.

```
#include <stdio.h>

// Function to return the maximum of two numbers using pointers
int findMax(int *a, int *b) {
    if (*a > *b)
        return *a;
    else
        return *b;
}

int main() {
    int num1, num2, max;

    // Input two numbers
    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);

    // Call function using pointers
    max = findMax(&num1, &num2);

    // Display result
    printf("The maximum of %d and %d is %d\n", num1, num2, max);

    return 0;
}
```

Q.2 Write a C program to Find the Length of a String Without using Built-in Function

```
#include <stdio.h>

int main() {
    char str[100];
    int i = 0;

    // Accept string from user
    printf("Enter a string: ");
    scanf("%s", str);  // Accepts one word (no spaces)

    // Count characters until null character '\0' is found
    while (str[i] != '\0') {
        i++;
    }

    // Display length
    printf("Length of the string is: %d\n", i);

    return 0;
}
```

---------------------SLIP 13--------------------------------

Q.1 Write a program to accept three integers as command line arguments and find the minimum, maximum and average of these numbers. Display error message if wrong number of arguments are entered.

```
#include <stdio.h>
#include <stdlib.h>  // for atoi()

int main(int argc, char *argv[]) {
    // Check if exactly 3 arguments are provided (excluding program
name)
    if (argc != 4) {
        printf("Error: Please enter exactly 3 integer arguments.\n");
        printf("Usage: %s <num1> <num2> <num3>\n", argv[0]);
        return 1;
    }

    // Convert arguments from strings to integers
    int a = atoi(argv[1]);
    int b = atoi(argv[2]);
    int c = atoi(argv[3]);

    // Find minimum
    int min = a;
    if (b < min) min = b;
    if (c < min) min = c;

    // Find maximum
    int max = a;
    if (b > max) max = b;
    if (c > max) max = c;

    // Calculate average
    float avg = (a + b + c) / 3.0;

    // Display results
    printf("Minimum: %d\n", min);
    printf("Maximum: %d\n", max);
    printf("Average: %.2f\n", avg);

    return 0;
}
```

Note : To run program Type
./a.out 2 3 4

Q.2
Write a Menu driven program to perform following operations on string till user selects exit.
a) Concatenate
b) Length

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

int main() {
    int choice;
    char str1[100], str2[100], result[200];

    while (1) {
        // Display menu
        printf("\n--- String Operations Menu ---\n");
        printf("1. Concatenate\n");
        printf("2. Find Length\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        // Handle choices
        switch (choice) {
            case 1:
                // Concatenate strings
                printf("Enter first string: ");
                scanf("%s", str1);
                printf("Enter second string: ");
                scanf("%s", str2);
```

```c
                strcpy(result, str1);   // Copy first string to result
                strcat(result, str2);   // Concatenate second string

                printf("Concatenated string: %s\n", result);
                break;

            case 2:
                // Find length of a string
                printf("Enter a string: ");
                scanf("%s", str1);
                printf("Length of the string: %lu\n", strlen(str1));
                break;

            case 3:
                // Exit
                printf("Exiting program.\n");
                return 0;

            default:
                printf("Invalid choice! Please try again.\n");
        }
    }

    return 0;
}
```

---SLIP 16 ---------------

Q. 1 Write a C program to read two integers using pointer and perform all arithmetic operations on them.

```
#include <stdio.h>

int main() {
    int num1, num2;
    int *ptr1, *ptr2;

    // Pointers point to the addresses of num1 and num2
    ptr1 = &num1;
    ptr2 = &num2;

    // Input two integers using pointers
    printf("Enter first integer: ");
    scanf("%d", ptr1);

    printf("Enter second integer: ");
    scanf("%d", ptr2);

    // Perform and display arithmetic operations
    printf("\n--- Arithmetic Operations ---\n");
    printf("Addition: %d + %d = %d\n", *ptr1, *ptr2, (*ptr1 + *ptr2));
    printf("Subtraction: %d - %d = %d\n", *ptr1, *ptr2, (*ptr1 -
*ptr2));
    printf("Multiplication: %d * %d = %d\n", *ptr1, *ptr2, (*ptr1 *
*ptr2));

    // Check for division by zero
    if (*ptr2 != 0) {
        printf("Division: %d / %d = %d\n", *ptr1, *ptr2, (*ptr1 /
*ptr2));
        printf("Modulus: %d %% %d = %d\n", *ptr1, *ptr2, (*ptr1 %
*ptr2));
    } else {
        printf("Division and Modulus by zero are not allowed.\n");
    }
    return 0;
}
```

Q.2 Write a C program which accepts a string and two characters as command line arguments and replace all occurrences of the first character by the second character.

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

int main(int argc, char *argv[]) {
    // Check if enough arguments are passed
    if (argc != 4) {
        printf("Usage: %s <string> <char_to_replace>
<replacement_char>\n", argv[0]);
        return 1;
    }

    char *str = argv[1];
    char ch1 = argv[2][0]; // character to be replaced
    char ch2 = argv[3][0]; // replacement character

    // Replace all occurrences of ch1 with ch2
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == ch1) {
            str[i] = ch2;
        }
    }

    // Display the modified string
    printf("Modified string: %s\n", str);

    return 0;
}
```

--SLIP 19-----------

Q1. Write C Program to find maximum of three numbers using nested macros.

```
#include <stdio.h>

// Macro to find maximum of two numbers
#define MAX2(a, b) ((a) > (b) ? (a) : (b))

// Nested macro to find maximum of three numbers
#define MAX3(a, b, c) (MAX2(MAX2(a, b), c))

int main() {
    int num1, num2, num3, max;

    // Input three numbers
    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    // Use nested macro to find maximum
    max = MAX3(num1, num2, num3);

   // Display the result
    printf("The maximum of %d, %d, and %d is %d\n", num1, num2, num3, 
max);

    return 0;
}
```

Q.2 Write a program to pass two strings to user defined function and copy one string to another using pointer.

```
#include <stdio.h>

// Function to copy string src to dest using pointers
void stringCopy(char *dest, const char *src) {
    while (*src != '\0') {
        *dest = *src;  // Copy character
        dest++;
        src++;
    }
    *dest = '\0';  // Null terminate destination string
}

int main() {
    char str1[100], str2[100];

    // Input first string
    printf("Enter first string: ");
    scanf("%s", str1);

    // Input second string (can be empty, it will be overwritten)
    printf("Enter second string (will be overwritten): ");
    scanf("%s", str2);

    // Copy str1 to str2 using user-defined function
    stringCopy(str2, str1);

    // Display copied string
    printf("After copying, second string is: %s\n", str2);

    return 0;
}
```

---------------------SLIP 20---

Q. 1 Write a program to display reverse of string using pointer.

```
#include <stdio.h>

int main() {
    char str[100];
    char *ptr;
    int length = 0;

    // Input string (no spaces)
    printf("Enter a string: ");
    scanf("%s", str);

    // Calculate length of string using pointer
    ptr = str;
    while (*ptr != '\0') {
        length++;
        ptr++;
    }

    // Set pointer to last character of string
    ptr = str + length - 1;

    // Print string in reverse using pointer
    printf("Reversed string: ");
    for (int i = 0; i < length; i++) {
        printf("%c", *ptr);
        ptr--;
    }
    printf("\n");

    return 0;
}
```

Q.2 Write a Menu driven program to perform following operation on the string:

a) To find total number of Vowels and consonants in it.
b) To check palindrome or not

```
#include <stdio.h>
#include <ctype.h>  // for tolower()

// Function to count vowels and consonants
void countVowelsConsonants(char str[], int *vowels, int *consonants) {
    *vowels = 0;
    *consonants = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        char ch = tolower(str[i]);
        if ((ch >= 'a' && ch <= 'z')) {
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch
== 'u') {
                (*vowels)++;
            } else {
                (*consonants)++;
            }
        }
    }
}

// Function to check palindrome
int isPalindrome(char str[]) {
    int start = 0;
    int end = 0;

    // Calculate length
    while (str[end] != '\0') {
        end++;
    }
    end--;  // Last valid character index

    // Check palindrome ignoring case
    while (start < end) {
        char chStart = tolower(str[start]);
        char chEnd = tolower(str[end]);
        if (chStart != chEnd)
            return 0;  // Not palindrome
        start++;
        end--;
    }
    return 1;  // Palindrome
}

int main() {
    char str[100];
    int choice;
    int vowels, consonants;

    while (1) {
        printf("\n--- String Operations Menu ---\n");
        printf("1. Count Vowels and Consonants\n");
        printf("2. Check Palindrome\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        // Clear input buffer before reading string
        getchar();

        switch (choice) {
            case 1:
                printf("Enter a string (no spaces): ");
                scanf("%s", str);
                countVowelsConsonants(str, &vowels, &consonants);
                printf("Vowels: %d\nConsonants: %d\n", vowels,
consonants);
                break;

            case 2:
                printf("Enter a string (no spaces): ");
                scanf("%s", str);
                if (isPalindrome(str)) {
                    printf("The string is a palindrome.\n");
                } else {
                    printf("The string is not a palindrome.\n");
                }
                break;

            case 3:
                printf("Exiting program.\n");
                return 0;

            default:
                printf("Invalid choice! Please try again.\n");
        }
    }
}
```

------------------SLIP 21 ---

Q.1 Write a c program to accept a string and store it in an array. Display the length of that string without using built in function.

```
#include <stdio.h>

int main() {
    char str[100];
    int length = 0;

    printf("Enter a string (no spaces): ");
    scanf("%s", str);

    // Count length manually
    while (str[length] != '\0') {
        length++;
    }

    printf("Length of the string is: %d\n", length);

    return 0;
}
```

Q.2 Write a C program to sort given array in ascending/descending order using pointer.

```
#include <stdio.h>

// Function to swap two integers using pointers
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Function to sort array in ascending order using pointers
```

```
void sortAscending(int *arr, int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (*(arr + i) > *(arr + j)) {
                swap(arr + i, arr + j);
            }
        }
    }
}

int main() {
    int n;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Sort array in ascending order
    sortAscending(arr, n);

    printf("Sorted array in ascending order:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", *(arr + i));
    }
    printf("\n");

    return 0;
}
```

-------------------SLIP 23 ---

Q.1 Write a program to find minimum of two integers using pointer.

```
#include <stdio.h>

int main() {
    int a, b;
    int *p1 = &a, *p2 = &b;

    printf("Enter two integers: ");
    scanf("%d %d", p1, p2);

    if (*p1 < *p2)
        printf("Minimum is %d\n", *p1);
    else
        printf("Minimum is %d\n", *p2);

    return 0;
}
```

Q.2 Write a C program to read two strings. If first string is greater than second then concatenate second to first and display concatenated string, If first string is smaller than second then concatenate first to second, otherwise display length of string. (Hint: use strcmp function)

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

int main() {
    char str1[100], str2[100];

    printf("Enter first string: ");
    scanf("%s", str1);

    printf("Enter second string: ");
    scanf("%s", str2);
    int cmp = strcmp(str1, str2);

    if (cmp > 0) {
        // str1 > str2: concatenate str2 to str1
        strcat(str1, str2);
        printf("Concatenated string: %s\n", str1);
    } else if (cmp < 0) {
        // str1 < str2: concatenate str1 to str2
        strcat(str2, str1);
        printf("Concatenated string: %s\n", str2);
    } else {
        // strings are equal, print length
        printf("Strings are equal. Length: %lu\n", strlen(str1));
    }

    return 0;
}
```

---------------------------SLIP 24---------------------------------

Q.1 Write a C program to find square of number using macro.

```
#include <stdio.h>

// Macro to calculate square of a number
#define SQUARE(x) ((x) * (x))

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Square of %d is %d\n", num, SQUARE(num));

    return 0;
}
```

Q.2 Write a C program to accept a string and one character from user. Check whether the given character is present in the given string or not? Display appropriate message. (Hint: use strchr or strrchr)

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

int main() {
    char str[100], ch;

    printf("Enter a string (no spaces): ");
    scanf("%s", str);

    printf("Enter a character to search: ");
    scanf(" %c", &ch);  // space before %c to skip any leftover
newline

    if (strchr(str, ch) != NULL) {
        printf("Character '%c' is present in the string.\n", ch);
    } else {
        printf("Character '%c' is NOT present in the string.\n", ch);
    }

    return 0;
}
```

---------------------------------SLIP 27-----

Q.1 Write a C program to display current time, date and name of file using predefined macros.

```
#include <stdio.h>

int main() {
    printf("Current Time: %s\n", __TIME__);
    printf("Current Date: %s\n", __DATE__);
    printf("File Name: %s\n", __FILE__);

    return 0;
}
```

Q.2 Write a C program to count the total number of uppercase and lowercase characters in a string.

```
#include <stdio.h>

int main() {
    char str[100];
    int uppercase = 0, lowercase = 0;

    printf("Enter a string (no spaces): ");
    scanf("%s", str);

    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] >= 'A' && str[i] <= 'Z')
            uppercase++;
        else if (str[i] >= 'a' && str[i] <= 'z')
            lowercase++;
    }

    printf("Uppercase letters: %d\n", uppercase);
    printf("Lowercase letters: %d\n", lowercase);
    return 0;
}
```

----------------------------SLIP 28-----------------------------------

Q.1 Write a C function to compare two string using pointer.

```
#include <stdio.h>

int compareStrings(char *s1, char *s2) {
    while (*s1 != '\0' && *s2 != '\0') {
        if (*s1 != *s2)
            break;
        s1++;
        s2++;
    }
    return (*s1 - *s2);
}

int main() {
    char str1[100], str2[100];

    printf("Enter first string: ");
    scanf("%s", str1);

    printf("Enter second string: ");
    scanf("%s", str2);

    int res = compareStrings(str1, str2);

    if (res == 0)
        printf("Strings are equal.\n");
    else if (res < 0)
        printf("First string is smaller.\n");
    else
        printf("First string is greater.\n");

    return 0;
}
```

Q.2 Write a C program to find sum and average of numbers given in command line arguments.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc < 2) {
        printf("Usage: %s num1 num2 ...\n", argv[0]);
        return 1;
    }

    int sum = 0;
    for (int i = 1; i < argc; i++) {
        sum += atoi(argv[i]);
    }

    double average = (double)sum / (argc - 1);

    printf("Sum = %d\n", sum);
    printf("Average = %.2f\n", average);

    return 0;
}
```

------------------------------------SLIP 30 -------

Q.1 Write a program to find sum of n elements entered by user (Hint: Use malloc() function to accept elements)

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n;
    int *arr;
    int sum = 0;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Allocate memory dynamically for n integers
    arr = (int *)malloc(n * sizeof(int));

    if (arr == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    // Input elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Calculate sum
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }

    printf("Sum of elements = %d\n", sum);
```

```c
    // Free the allocated memory
    free(arr);

    return 0;
}
```

Q.2 Write a menu driven C program to perform the following operations on strings using standard library functions: a) Copy the contents of one string to another string b) Convert the string to uppercase string

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

int main() {
    char str1[100], str2[100];
    int choice;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Copy string\n");
        printf("2. Convert string to uppercase\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        getchar();  // Consume newline character left in input buffer

        switch (choice) {
            case 1:
                printf("Enter the source string: ");
                fgets(str1, sizeof(str1), stdin);
                str1[strcspn(str1, "\n")] = '\0';  // Remove trailing
newline

                // Copy str1 to str2
                strcpy(str2, str1);

                printf("Copied string: %s\n", str2);
                break;
            case 2:
                printf("Enter the string to convert to uppercase: ");
                fgets(str1, sizeof(str1), stdin);
                str1[strcspn(str1, "\n")] = '\0';  // Remove trailing
newline

                for (int i = 0; str1[i] != '\0'; i++) {
                    str1[i] = toupper((unsigned char)str1[i]);
                }

                printf("Uppercase string: %s\n", str1);
                break;

            case 3:
                printf("Exiting program.\n");
                return 0;

            default:
                printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```