**Assignment 9**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void reverseString(char\* str);

// Custom function for strrev (not standard in string.h)

void reverseString(char\* str) {

    int len = strlen(str);

    for (int i = 0; i < len / 2; i++) {

        char temp = str[i];

        str[i] = str[len - 1 - i];

        str[len - 1 - i] = temp;

    }

}

int main() {

    char str1[100] = "Hello";

    char str2[100] = "World";

    char str3[100];

    printf("Original str1: %s\n", str1);

    printf("Original str2: %s\n\n", str2);

    // 1. strlen

    printf("Length of str1: %zu\n", strlen(str1));

    // 2. strcpy

    strcpy(str3, str1);

    printf("Copy str1 to str3: %s\n", str3);

    // 3. strncpy

    strncpy(str3, str2, 3);

    str3[3] = '\0';  // null terminate manually

    printf("Copy first 3 chars of str2 to str3: %s\n", str3);

    // 4. strcat

    strcat(str1, str2);

    printf("Concatenate str2 to str1: %s\n", str1);

    // 5. strncat

    strncat(str1, "!!", 2);

    printf("Concatenate '!!' to str1: %s\n", str1);

    // 6. strcmp

    printf("Compare str1 and str2: %d\n", strcmp(str1, str2));

    // 7. strncmp

    printf("Compare first 3 chars of str1 and str2: %d\n", strncmp(str1, str2, 3));

    // 8. strchr

    printf("First occurrence of 'l' in str1: %s\n", strchr(str1, 'l'));

    // 9. strrchr

    printf("Last occurrence of 'l' in str1: %s\n", strrchr(str1, 'l'));

    // 10. strstr

    printf("Find substring 'lo' in str1: %s\n", strstr(str1, "lo"));

    // 11. strpbrk

    printf("First match of any char from 'aeiou' in str1: %s\n", strpbrk(str1, "aeiou"));

    // 12. strspn

    printf("Length of initial segment in str1 containing only vowels: %zu\n", strspn(str1, "aeiou"));

    // 13. strcspn

    printf("Length of initial segment in str1 without vowels: %zu\n", strcspn(str1, "aeiou"));

    // 14. strtok

    char temp[] = "Hi-Manish-Here";

    char\* token = strtok(temp, "-");

    printf("Tokens:\n");

    while (token != NULL) {

        printf("  %s\n", token);

        token = strtok(NULL, "-");

    }

    // 15. reverseString (custom strrev)

    char revStr[] = "ReverseMe";

    reverseString(revStr);

    printf("Reversed string: %s\n", revStr);

    // 16. memset

    char buffer[10];

    memset(buffer, '\*', 5);

    buffer[5] = '\0';

    printf("After memset: %s\n", buffer);

    // 17. memcpy

    char src[] = "CopyThis";

    char dest[20];

    memcpy(dest, src, strlen(src)+1);

    printf("Copied string using memcpy: %s\n", dest);

    // 18. memmove (overlapping copy)

    char text[] = "abcdefgh";

    memmove(text + 2, text, 5);

    text[7] = '\0';

    printf("After memmove with overlap: %s\n", text);

    // 19. memcmp

    printf("Memory compare: %d\n", memcmp("abc", "abc", 3));

    // 20. isalpha, isdigit, isspace demo

    char c = 'M';

    printf("isalpha('%c'): %d\n", c, isalpha(c));

    c = '5';

    printf("isdigit('%c'): %d\n", c, isdigit(c));

    c = ' ';

    printf("isspace(' '): %d\n", isspace(c));

    // 21. toupper / tolower

    char ch = 'g';

    printf("toupper('%c'): %c\n", ch, toupper(ch));

    ch = 'G';

    printf("tolower('%c'): %c\n", ch, tolower(ch));

    return 0;

}