LAB ASSIGNENT 3

AMISH PRIYADARSHI

102269002

1). #include<stdio.h>

```
//write     a program to implement strlen() function
#include <stdio.h>


size_t strlen_custom(const char *str) {
        size_t length = 0;
        while (*str != '\0') {
                length++;
                str++;
        }
        return length;
}


int main() {
        char myString[] = "XXXTENTACTION,18/06 !";
```

```
    size_t length = strlen_custom(myString);

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

    return 0;
}
```

2).

```
#include <stdio.h>
//write a program to implement strcpy() function

char *strcpy_custom(char *dest, const char *src) {
    char *originalDest = dest; // Store the original destination pointer

    while (*src != '\0') {
        *dest = *src;
        dest++;
        src++;
    }
```

```
    *dest = '\0'; // Null-terminate the destination string
    return originalDest;
}

int main() {
    char source[] = "KANYE ,MADE HER FAMOUS!";
    char destination[20]; // Make sure the destination array is large enough

    strcpy_custom(destination, source);

    printf("Source: %s\n", source);
    printf("Destination: %s\n", destination);

    return 0;
}
```

3).

```
#include <stdio.h>
//write a program to implement strcat() function

char *strcat_custom(char *dest, const char *src) {
    char *originalDest = dest; // Store the original destination pointer
```

```
    // Move dest pointer to the end of the string
    while (*dest != '\0') {
        dest++;
    }

    // Copy characters from src to the end of dest
    while (*src != '\0') {
        *dest = *src;
        dest++;
        src++;
    }

    *dest = '\0'; // Null-terminate the concatenated string
    return originalDest;
}

int main() {
    char destination[30] = "TREACHEROUS , ";
    char source[] = "TWINS!";

    strcat_custom(destination, source);
```

```
    printf("Concatenated string: %s\n", destination);

    return 0;
}
```

4).

```
#include <stdio.h>

//WAP to implement strcmp() function

int strcmp_custom(const char *str1, const char *str2) {
    while (*str1 != '\0' || *str2 != '\0') {
        if (*str1 != *str2) {
            return (*str1 - *str2);
        }
        str1++;
        str2++;
    }
    return 0;
}
```

```
int main() {
	char string1[] = "UTOPIA";
	char string2[] = "ASTROWORLD";

	int result = strcmp_custom(string1, string2);

	if (result < 0) {
		printf("String 1 is less than String 2\n");
	} else if (result > 0) {
		printf("String 1 is greater than String 2\n");
	} else {
		printf("String 1 is equal to String 2\n");
	}

	return 0;
}
```

5). #include <stdio.h>

//WAP to demostrate limitations of two-dimentional array of characters.

```
int main() {
	char names[3][10] = {
		"TRAVIS",
		"DRAVE",
		"KENDRICK"
	};

	// Display the names
	for (int i = 0; i < 3; i++) {
		printf("Name %d: %s\n", i + 1, names[i]);
	}

	// Attempt to modify a name
	names[1][0] = 'A'; // Attempting to change "DRAKE" to "ARAKE"

	// Display the names again
	for (int i = 0; i < 3; i++) {
		printf("Name %d: %s\n", i + 1, names[i]);
	}

	return 0;
}
```

6).#include <stdio.h>

//WAP to demostrate an array of pointers to strings.

```
int main() {
	char *names[] = {
		"METROOOOOOOOOOOOOOO",
		"SAVAGE",
		"A$AP"
	};

	// Display the names using the array of pointers
	for (int i = 0; i < 3; i++) {
		printf("Name %d: %s\n", i + 1, names[i]);
	}

	// Modify a name using the pointer
	names[1] = "TUPAC"; // Changing "SAVAGE" to "TUPAC"

	// Display the modified names
```

```
    for (int i = 0; i < 3; i++) {
        printf("Name %d: %s\n", i + 1, names[i]);
    }

    return 0;
}
```