Assignment 10

a.mystrcpy

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

void main(){

        char src[]="Hello,World";
        char dest[30];

        my_strcpy(dest,src);

        printf("Source:%s\n",src);
        printf("Copied:%s\n",dest);
}

my_strcpy(char *dest , const char *src){

        int i=0;
        while(src[i]!='\0'){
                dest[i]=src[i];
                i++;
        }

        dest[i]='\0';
}
```

b. mystrlen

```
#include<stdio.h>

void main(){

        char str[10]="firstbit";

         int l=callength(str);
         printf("%d",l);
}

int callength(char *ptr){

        int i=0;
        while(ptr[i]!='\0'){
                i++;

        }
        return i;
}
```

c. mystrcmp

```
#include<stdio.h>
int mystrcmp();

void main(){

        char str1[]="Hello";
        char str2[]="Hello";

        int result=mystrcmp(str1,str2);

        if(result<0){

                printf("String 1 is samller.");

        }
        else if(result>0){

                printf("String 1 greater.");
        }

        else
         printf("Both strings are equal.");



}

 int mystrcmp(char *str1, char *str2){

        int i=0;
```

```
        while(str1[i]!='\0' && str2[i]!='\0'){

                if(str1[i]<str2[i]){

                        return -1;

                }
                else if(str1[i]>str2[i]){

                        return 1;
                }

                        i++;

        }
         if(str1[i] == '\0' && str2[i] == '\0') return 0;
   if(str1[i] == '\0') return -1;
   return 1;

}
```

d. mystrcat

```
#include<stdio.h>

char *mystrcat(char *str1,char *str2);

void main(){
```

```
	char str1[]="Hello";
	char str2[]="World";

	mystrcat(str1 ,str2);

	printf("%s\n",str1);
}


char *mystrcat(char *str1,char *str2){

	int i=0,j=0;

	while(str1[i]!='\0'){

		i++;


	}


	while(str2[j]!='\0'){

		str1[i]=str2[j];

		i++;
		j++;
	}

	return str1;
}
```

e.mystrncpy

```
#include <stdio.h>



void main() {
    char src[] = "Hello, World";
    char dest[30];

    my_strncpy(dest, src, 6);

    printf("Source: %s\n", src);
    printf("Copied: %s\n", dest);
}



void my_strncpy(char *dest,  char *src, size_t n) {
    size_t i = 0;
    while (i < n && src[i] != '\0') {
        dest[i] = src[i];
        i++;
    }
    while (i < n) {
        dest[i++] = '\0';
    }
}
```

f. mystrupper

```
#include <stdio.h>

void my_strupper(char *str) {
   int i = 0;
   while (str[i] != '\0') {
      str[i] = toupper((unsigned char) str[i]);
      i++;
   }
}

void main() {
   char str[] = "Hello, World";

   my_strupper(str);

   printf("Uppercase: %s\n", str);
}
```

g.mystrlower

```
#include <stdio.h>
void main() {
   char str[] = "Hello, WORLD";

   my_strlower(str);

   printf("Lowercase: %s\n", str);
}
void my_strlower(char *str) {
```

```
    int i = 0;
    while (str[i] != '\0') {
        str[i] = tolower((unsigned char) str[i]);
        i++;
    }
}
```

h.mystrrev

```
#include <stdio.h>
void main() {
    char str[] = "Hello, World";

    my_strrev(str);

    printf("Reversed: %s\n", str);
}



void my_strrev(char *str) {
            int i;
    int len = strlen(str);
    for ( i = 0; i < len / 2; i++) {
        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}
```

i. mystrstr

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

char* mystrstr(const char* str, const char* substr) {
    if (!*substr) return (char*)str;

    while (*str) {
        const char *s = str, *sub = substr;
        while (*s && *sub && *s == *sub) {
            s++;
            sub++;
        }
        if (!*sub) return (char*)str;
        str++;
    }
    return NULL;
}

void main() {
    char str[] = "hello world";
    char substr[] = "world";

    char *result = mystrstr(str, substr);
    if (result)
        printf("Substring found at index %ld\n", result - str);
    else
        printf("Substring not found\n");

}
```

j.mystrcasecmp

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

int mystrcasecmp(const char* str1, const char* str2) {
    while (*str1 && *str2) {
        char c1 = tolower((unsigned char)*str1);
        char c2 = tolower((unsigned char)*str2);

        if (c1 != c2) return c1 - c2;

        str1++;
        str2++;
    }
    return tolower((unsigned char)*str1) - tolower((unsigned char)*str2);
}

int main() {
    char str1[] = "Hello";
    char str2[] = "hello";

    int result = mystrcasecmp(str1, str2);

    if (result == 0)
        printf("Strings are equal (case insensitive)\n");
    else
        printf("Strings are different\n");

    return 0;
}
```

k. mystrchr

```
#include<stdio.h>
char *mystrchr(char *str,char ch);
void main(){

	char str[10]="Welcomehome";
	char ch='m';
	char *result=mystrchr(str,ch);
	if(result!=NULL){
		printf("Character '%c' found at index %d",ch,result-str);
	}
	else
	printf("Character not found.");
}
char *mystrchr(char *str,char ch){
	int i=0;
	while(str[i]!='\0'){
		if(str[i]==ch){
					return (str+i);
		}



		i++;
	}

		return NULL;

}
```

I. mystrrchr

```
#include <stdio.h>
char *mystrrchr(char *str, char ch);
void main() {
   char str[] = "Welcomehome";
   char ch = 'm';
   char *result = mystrrchr(str, ch);
   if (result != NULL) {
      printf("Last occurrence of '%c' found at index %d\n", ch, result - str);
   } else {
      printf("Character not found.\n");
   }
}
char *mystrrchr(char *str, char ch) {
   char *last_occurrence = NULL;
   int i = 0;
   while (str[i] != '\0') {
      if (str[i] == ch) {

         last_occurrence = str+i;

          }
      i++;
   }

   return last_occurrence;
}
```

m. mystrncmp

```
#include <stdio.h>
int mystrncmp(char *str1, char *str2, int n);

void main() {
   char str1[] = "Hello";
   char str2[] = "Helium";
   int n = 3;

   int result = mystrncmp(str1, str2, n);

   if (result == 0) {
      printf("First %d characters of both strings are equal.\n", n);
   } else if (result < 0) {
      printf("First %d characters of \"%s\" are less than \"%s\".\n", n, str1, str2);
   } else {
      printf("First %d characters of \"%s\" are greater than \"%s\".\n", n, str1, str2);
   }
}

int mystrncmp(char *str1,char *str2, int n) {
   int i = 0;
   while (i < n && str1[i] != '\0' && str2[i] != '\0') {
      if (str1[i] != str2[i]) {
         return str1[i] - str2[i];
      }
      i++;
   }
```

```
    if (i < n) {
        return str1[i] - str2[i];
    }

    return 0;
}
```

n.mystrnstr

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

char* mystrnstr(const char* str, const char* substr, size_t n) {
    size_t sub_len = strlen(substr);

    if (sub_len == 0) return (char*)str;
    size_t  i;

    for (i = 0; i <= n - sub_len && str[i] != '\0'; i++) {
        if (strncmp(&str[i], substr, sub_len) == 0) {
            return (char*)&str[i];
        }
    }

    return NULL;
}

int main() {
    char str[] = "hello world";
    char substr[] = "wor";
```

```
    char *result = mystrnstr(str, substr, 8);

    if (result)
        printf("Substring found at index %ld\n", result - str);
    else
        printf("Substring not found within given range\n");

    return 0;
}
```

o.mystrncat

```
#include <stdio.h>

char *mystrncat(char *dest, char *src, int n);

void main() {
    char dest[20] = "Hello, ";
    char src[] = "World!";
    int n = 5;

    mystrncat(dest, src, n);

    printf("Concatenated String: %s\n", dest);
}

char *mystrncat(char *dest,char *src, int n) {
    int i = 0, j = 0;
```

```
    while (j < n && src[j] != '\0') {
        dest[i] = src[j];
        i++;
        j++;
    }


    dest[i] = '\0';

    return dest;
}
```

p.mystrncasecmp

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

int mystrncasecmp(const char* str1, const char* str2, size_t n) {

            size_t i;
    for ( i = 0; i < n && (str1[i] != '\0' || str2[i] != '\0'); i++) {
        char c1 = tolower((unsigned char)str1[i]);
        char c2 = tolower((unsigned char)str2[i]);

        if (c1 != c2) return c1 - c2;
    }
    return 0; // Strings are equal up to `n` characters
}

int main() {
    char str1[] = "HelloWorld";
```

```
    char str2[] = "helloworLD";

    int result = mystrncasecmp(str1, str2, 5); // Compare first 5 characters case-insensitively

    if (result == 0)
        printf("Strings are equal (case insensitive, first 5 chars)\n");
    else
        printf("Strings are different\n");

    return 0;
}
```