Assignment 9

1. //memcpy()

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

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

    memcpy(dest, src, strlen(src) + 1);
    printf("Copied string: %s\n", dest);

    return 0;
}
```

2. //memmove

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

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

    // Overlapping memory regions: moving part of the string within itself
    memmove(str + 7, str, 5);
    printf("After memmove: %s\n", str);

    return 0;
}
```

3. //memset()

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

int main() {
    char buffer[20];

    memset(buffer, '*', 10);  // Fill first 10 bytes with '*'
    buffer[10] = '\0';        // Null terminate for printing
    printf("Buffer after memset: %s\n", buffer);

    return 0;
}
```

4. //sprintf()

```
#include <stdio.h>

int main() {
    char buffer[100];
    int age = 25;
    float height = 5.9;

    sprintf(buffer, "Age: %d, Height: %.1f feet", age, height);

    printf("Formatted string: %s\n", buffer);

    return 0;
}
```

5. //strcat() function is used to concatenate (append) one string to another. It appends the second string

```
//(src) to the end of the first string (dest) and returns dest.
int main() {
    char dest[50] = "Hello, ";
    char src[] = "World!";

    strcat(dest, src);

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

}
```

6. //strchr()

```
#include <stdio.h>
#include <string.h>
int main(){
        char myStr[] = "Hello World";
char *myPtr = strchr(myStr, 'W');
if (myPtr != NULL) {
  printf("%s", myPtr);
}
else{
        printf("Character not found");
}
}
```

7. //#include <stdio.h>

```
#include <string.h>

int main() {
    char str1[] = "apple";
    char str2[] = "banana";
    char str3[] = "apple";

    printf("Comparing str1 and str2: %d\n", strcmp(str1, str2)); // Negative value
```

```
   printf("Comparing str1 and str3: %d\n", strcmp(str1, str3)); // 0
   printf("Comparing str2 and str1: %d\n", strcmp(str2, str1)); // Positive value

}
```

8. //The strcpy() function copies a string from one location to another, including the null terminator

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

int main() {
   char src[] = "Hello, World!";
   char dest[50];

   strcpy(dest, src);

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

}
```

9. //strcspn()

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

int main() {
   char str1[] = "hello world";
   char str2[] = "owd";  // Searching for 'o', 'w', or 'd' in str1

   size_t length = strcspn(str1, str2);

   printf("Initial segment length before any match: %zu\n", length);

   return 0;
}
```

10.//strdup() used to duplicate a string by allocating memory for a new copy and copying the original

```
//string into it
int main()
{
        char original[] = "Hello World!";

        char *duplicate = strdup(original);
        printf("%s", duplicate);
}
```

11.//strlen():- finds the length of the string excludinh '\0' character

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

```
#include <string.h>
int main()
{
        char str[] = "Hello World!";
        int len = strlen(str);
        printf("%d",len);
}
    12.//strncat() function in C is used to concatenate (append) a specified number of
        characters from one
//string to another.
#include <stdio.h>
#include <string.h>

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

   strncat(dest, src, 3);
   printf("%s\n", dest);
}
    13.//strncpy():- copies n characters from source to the destination
#include <stdio.h>
#include <string.h>
int main()
{
        char src[] = "Hello World";
        char dest[20] = "Code";

        strncpy(dest, src,6);
        printf("%s", dest);
}
    14.//strpbrk()
#include <stdio.h>
#include <string.h>

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

   char *ptr = strpbrk(str1, str2);

   if (ptr != NULL) {
      printf("First matching character: '%c' found at position: %ld\n", *ptr, ptr - str1);
   } else {
        printf("No matching characters found.\n");
    }

    return 0;
}
```

15.//strrchr()

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

int main() {
    char str[] = "Hello, world!";
    char ch = 'o';

    char *ptr = strrchr(str, ch);

    if (ptr != NULL) {
        printf("Last occurrence of '%c' found at position: %ld\n", ch, ptr - str);
    } else {
        printf("Character not found.\n");
    }
}
```

16.//strspn()

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

int main() {
    char str1[] = "123456ABC";
    char str2[] = "1234567890";

    size_t length = strspn(str1, str2);

    printf("Length of initial matching segment: %zu\n", length);

    return 0;
}
```

17.//strstr()

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

int main() {
    char str[] = "Hello, world!";
    char sub[] = "world";

    char *ptr = strstr(str, sub);
    if (ptr != NULL) {
        printf("Substring found at position: %ld\n", ptr - str);
    } else {
        printf("Substring not found.\n");
    }

}
     18.//strtok()
#include <stdio.h>
#include <string.h>

int main() {
    char str[] = "Hello, World! Welcome to C.";
    char *token = strtok(str, " ,.!");

    while (token != NULL) {
        printf("%s\n", token);
        token = strtok(NULL, " ,.!");
    }

    return 0;
}
```