Q.Use around 21 built-in functions of string in program

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

int main() {

   //strlen
   char str[] = "Ajit";
   printf("%s ", str);
   int len = strlen(str);
   printf("\nLength of string is :%d", len);

   //strcpy
   char str1[10]="Ajit";
   char str2[10];
   char* a= strcpy(str2,str1);
   printf("%s",a);
   printf("\n%s ",str2);

   //strncpy
   char str1[10]="Ajit";
   char str2[10];
   char* a= strncpy(str2,str1,2);
   str2[2]='\0';
   printf("%s",a);
   printf("\n%s ",str2);

   //strcat
   char str1[10]=" Ajit";
   char str2[15]="Chaudhari";
   char* a= strcat(str2,str1);
   printf("%s",a);
```

```
//strncat
char str1[10]=" Ajit";
char str2[15]="Chaudhari";
char* a= strncat(str2,str1,3);
printf("%s",a);

//strcmp
char str1[10]="Ajit";
char str2[15]="Ajit";
int result= strcmp(str2,str1);
if (result == 0)
   printf("Strings are equal\n");
else if (result > 0)
   printf("str2 is greater\n");
else
   printf("str1 is greater\n");

//strncmp
char str1[10]="Ajit";
char str2[15]="Ajit chaudhari";
int result= strncmp(str2,str1,2);
if (result == 0)
   printf("Strings are equal\n");
else
   printf("Strings are not equal\n");

//strchr
char str1[10]="Ajit";
char str2='a';
char* result= strchr(str1,str2);
if (result != NULL)
   printf("Character '%c' found at position: %ld\n", str2, result - str1);
else
```

```
    printf("Character '%c' not found\n", str2);

//strrchr
char str1[10]="Ajtit";
char str2='t';
char* result= strrchr(str1,str2);
if (result != NULL)
    printf("Character '%c' found at position: %ld\n", str2, result - str1);
else
    printf("Character '%c' not found\n", str2);



// strstr
char str1[] = "Hello, welcome to C programming!";
char str2[] = "C";
char* result = strstr(str1, str2);
if (result != NULL)
    printf("Substring found at position: %ld\n", result - str1);
else
    printf("Substring not found\n");

 //strspn
 char str1[] = "Hello, welcome to C programming!";
 char str2[] = "Hello";
 int result = strspn(str1, str2);
 printf("length of substring : %d\n", result);

//strcspn
char str1[] = "Hello, welcome to C programming!";
char str2[] = "C";
int result = strcspn(str1, str2);
printf("First character NOT in str2 found at position: %d\n", result);

//strpbrk
```

```
char str1[] = "Hello, welcome to C programming!";
char str2[] = "C";
char* result = strpbrk(str1, str2);
if (result != NULL)
    printf("First occurrence found at position: %d\n", result - str1);
else
    printf("Character not found\n");

//strtok
char str[] = "Hello, welcome to C programming!";
const char delim[] = " ,!";  // Delimiters: space, comma, exclamation mark
char *token;
token = strtok(str, delim);
while (token != NULL) {
    printf("Token: %s\n", token);
    token = strtok(NULL, delim); // Get the next token
}

//memset
char str[20] = "Hello World!";
memset(str, '#',5);
printf("Modified string %s",str);

//memcpy
char src[]="Ajit Chaudhari";
char dest[20];
memcpy(dest, src,strlen(src)+1);
printf("Copied string : %s",dest);

//memmove
char str[] = "Hello, World!";
memmove(str + 7, str, 5);  // Overlapping regions handled safely
printf("Modified string: %s\n", str);
```

```
    //memcmp
    char str1[] = "Hello";
    char str2[] = "Hello";
    char str3[] = "World";
    int result1 = memcmp(str1, str2, 5);  // Compare first 5 bytes
    int result2 = memcmp(str1, str3, 5);  // Compare first 5 bytes
    if (result1 == 0)
        printf("str1 and str2 are equal\n");
    else
        printf("str1 and str2 are different\n");
    if (result2 == 0)
        printf("str1 and str3 are equal\n");
    else
        printf("str1 and str3 are different\n");

    //memchr
    char str[] = "Hello, World!";
    char *ptr = memchr(str, 'W', strlen(str));
    if (ptr != NULL)
        printf("Found 'W' at position: %ld\n", ptr - str);
    else
        printf("'W' not found\n");

    return 0;
}
```