**strlen**
- **strlen(char *s)**
  - **Input:** Pointer to a string array
  - **Output:** Returns the number of chars in array until the null character
  - **Example:**
    - int N;
    - char s1[15] = “Hello”;
    - char s2[15] = “My name is:”;
    - N = strlen(s1); /* N is 5 */
    - N = strlen(s2); /* N is 11 */
    - N = strlen(“How are you?”); /* N is 12 */

**strcpy**
- **strcpy(char *dest, char *src)**
  - **Input:** Pointers to destination and source char arrays
  - **Output:** Copies src string into dest string
  - **Example:**
    - char s1[15] = “Hello”;
    - char s2[15] = “My name is:”;
    - char s3[40];
    - strcpy(s3, s1); /* s3 contains Hello\0 */
    - strcpy(s3, s2); /* s3 contains My Name is:\0 */
    - strcpy(s3, “How are you?”); /* s3 contains How are you?\0 */
**strncpy**

- `strncpy(char *dest, char *src, int n)`
  - **Input:** Pointers to `dest` and `src` char arrays, and an int
  - **Output:** Copies the first `n` chars into `dest` string; if the length of `src` is less than `n`, remaining space is padded with null char
  - **Example:**
    - `char s1[15] = "Hello";`
    - `char s2[15] = "My name is:";`
    - `char s3[40];`
    - `strncpy(s3, s1, 7); /* s3 contains Hello\0 */`
    - `strncpy(s3, s2, 7); /* s3 contains My Name */`
    - `strncpy(s3, "How are you?", 7); /* s3 contains How are */`

**strcat**

- `strcat(char *dest, char *src)`
  - **Input:** Pointers to destination and source char arrays
  - **Output:** Appends `src` string into `dest` string including the null pointer
  - **Example:**
    - `char s1[15] = "Hello";`
    - `char s2[6] = "John";`
    - `char s3[40];`
    - `strcpy(s3, s1); /* s3 contains Hello\0 */`
    - `strcat(s3, " "); /* s3 contains Hello \0 */`
    - `strcat(s3, s2); /* s3 contains Hello John\0 */`
strncat

- **strncat**(char *dest, char *src, int n)
  - **Input:** Pointers to dest and src char arrays, and an integer
  - **Output:** Appends the first n chars of the src string to the end of dest string; if the length of src is less than n, remaining space is padded with null char, if more, the first n chars and the null char are appended.

- **Example:**
  - char s1[15] = “Hello”;
  - char s3[40];
  - strcpy(s3, s1); /* s3 contains Hello\0 */
  - strcat(s3, “ “); /* s3 contains Hello \0 */
  - strncat(s3, s2, 6); /* s3 contains Hello John\0 \0 */
  - strncat(s3, s1, 3); /* s3 contains Hello JohnHel */

strncmp

- **strncmp**(char *s1, char *s2)
  - **Input:** Pointers to two strings
  - **Output:** compares s1 to s2 alphabetically and returns
    - 0 if identical,
    - a negative value if s1 comes before s2,
    - a positive value if s2 comes before s1

- **Example:**
  - int n;
  - char s1[15] = “Can”;
  - char s2[15] = “Caner”;
  - char s3[15] = “Cem”;
  - n = strcmp(s1, “Can”); /* n is zero */
  - n = strcmp(s1, s2); /* n is negative */
  - n = strcmp(s2, s1); /* n is positive */
  - n = strcmp(s1, s3); /* n is negative */
**strncpy**

- **strncpy(char *s1, char *s2, int n)**
  - **Input:** Pointers to two strings and an integer
  - **Output:** compares the first n chars of s1 to s2 alphabetically and returns
    - 0 if identical,
    - a negative value if s1 comes before s2,
    - a positive value if s2 comes before s1
  - **Example:**
    ```c
    int n;
    char s1[15] = "Can";
    char s2[15] = "Caner";
    char s3[15] = "Cem";
    n = strncpy(s1, s2, 4); /* n is negative */
    n = strncpy(s1, s2, 3); /* n is zero */
    n = strncpy(s1, s3, 3); /* n is negative */
    ```

**strtok**

- **strtok(char *s1, char *delim)**
  - **Input:** Pointers to two strings. After the first call, s1 is NULL
  - **Output:** Searches s1 for any of the characters in delim and replaces the first such character with \0. Subsequent calls to this function with NULL as the first argument, searches for more delimiter characters in s1 and replaces the first occurrence with \0. If a delimiter is found, strtok returns a pointer to the next character. If no delimiter is found, returns NULL.
  - **Example:**
    ```c
    char s1[] = "Can-Tan-Betul";
    strtok(s1, ",");
    strtok(NULL, ",");
    strtok(NULL, ",");
    ```
**strtok**

**Example:**
```c
char str[] = "Hearts#Spades#Diamonds#Clubs";
char delims[] = ";#";
char *result;
result = strtok(str, delims);
while(result != NULL) {
    printf("result is \"%s\n", result);
    result = strtok(NULL, delims);
}
```

**Output:**
```
result is "Hearts"
result is "Spades"
result is "Diamonds"
result is "Clubs"
```

---

**strtok**

**Example:**
```c
char str[] = "Hearts-Spades-Diamonds-Clubs";
char delims[] = ";-#";
char *result;
result = strtok(str, delims);
while(result != NULL) {
    printf("result is \"%s\n", result);
    result = strtok(NULL, delims);
}
```

**Output:**
```
result is "Hearts-"
result is "-Spades-"
result is "-Diamonds-"
result is "-Clubs"
```
### strtok

**Example:**

```c
char str[] = "Hearts-#-Spades-#-Diamonds-#-Clubs";
char delims[] = "-#";
char *result;
result = strtok(str, delims);
while(result != NULL) {
    printf("result is \"%s\n", result);
    result = strtok(NULL, delims);
}
```

**Output:**

- result is "Hearts"
- result is "Spades"
- result is "Diamonds"
- result is "Clubs"

---

### strtok

**Example:**

```c
char str[] = "January 15, 2010";
char delims[] = ",;";
char *result;
result = strtok(str, delims);
while(result != NULL) {
    printf("result is \"%s\n", result);
    result = strtok(NULL, delims);
}
```

**Output:**

- result is "January"
- result is "15"
- result is "2010"