1. Strlen( char S1[] ); Returns an integer equal to the length of S1. (The null character, ‘\0’, is not counted in the length.)
2. StrCpy( char S1[], char S2[]); Copies the S2 into the C-sting variable S1.
3. StrCat( char S1[], char S2[]); Concatenates the C string value of S2 onto the end of the C string in the C-string variable S1.
4. StrCmp(char S1[], char S2[]); Returns 0 if S1 and S2 are the same. Returns a value<0, if S1 is less than S2. Return s a value>0 if S1 is greater than S2( that is , returns a nonzero value if S1 and S2 are different.) The order is lexicographic.
5. StrChr( char S, int character);Returns a pointer to the first occurrence of character in the C string S. If the character is not found, the function returns a null pointer.
6. StrRChr( char S, int character); Returns a pointer to the last occurrence of character in the C sting S. If the character is not found, the function returns a null pointer. The terminating null-character is considered part of the C string. Therefore, it can also be located to retrieve a pointer to the end of a string.
7. StrNCpy( char S1, char S2, Limit); The same as the two-argument strcpy except that at most Limit characters are copied.
8. StrNCat( char S1, char S2, Limit); The same as the two-argument strcat except that at most Limit characters are appended.
9. FindAny(char S[ ], char charset[ ], int pos); Find the first occurrence of any of the characters in the charSet array in the source array. Return its positon.
10. FindNotAny(char S[ ], char charset[ ], intpos); Find the first occurrence of a character NOT in the characters of the charSet, in the source array. Return its position.
11. ToUpper(char S1[ ], char S2[ ]); Convert the C string to upper case.
12. ToLower(char S1[ ], char S2[ ]); Convert the C string to lower case.
13. GT, LT, EQ(char S1[ ], char S2[ ]); Boolean functions to compare two C strings.