|  |  |
| --- | --- |
| **Sr.No.** | **Function & Description** |
| 1 | [**void \*memchr(const void \*str, int c, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_memchr.htm)  Searches for the first occurrence of the character c (an unsigned char) in the first n bytes of the string pointed to, by the argument *str*. |
| 2 | [**int memcmp(const void \*str1, const void \*str2, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_memcmp.htm)  Compares the first n bytes of *str1* and *str2*. |
| 3 | [**void \*memcpy(void \*dest, const void \*src, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_memcpy.htm)  Copies n characters from src to *dest*. |
| 4 | [**void \*memmove(void \*dest, const void \*src, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_memmove.htm)  Another function to copy n characters from *str2* to *str1*. |
| 5 | [**void \*memset(void \*str, int c, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_memset.htm)  Copies the character c (an unsigned char) to the first n characters of the string pointed to, by the argument *str*. |
| 6 | [**char \*strcat(char \*dest, const char \*src)**](https://www.tutorialspoint.com/c_standard_library/c_function_strcat.htm)  Appends the string pointed to, by *src* to the end of the string pointed to by *dest*. |
| 7 | [**char \*strncat(char \*dest, const char \*src, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_strncat.htm)  Appends the string pointed to, by *src* to the end of the string pointed to, by *dest* up to n characters long. |
| 8 | [**char \*strchr(const char \*str, int c)**](https://www.tutorialspoint.com/c_standard_library/c_function_strchr.htm)  Searches for the first occurrence of the character c (an unsigned char) in the string pointed to, by the argument *str*. |
| 9 | [**int strcmp(const char \*str1, const char \*str2)**](https://www.tutorialspoint.com/c_standard_library/c_function_strcmp.htm)  Compares the string pointed to, by *str1* to the string pointed to by *str2*. |
| 10 | [**int strncmp(const char \*str1, const char \*str2, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_strncmp.htm)  Compares at most the first n bytes of *str1* and *str2*. |
| 11 | [**int strcoll(const char \*str1, const char \*str2)**](https://www.tutorialspoint.com/c_standard_library/c_function_strcoll.htm)  Compares string *str1* to *str2*. The result is dependent on the LC\_COLLATE setting of the location. |
| 12 | [**char \*strcpy(char \*dest, const char \*src)**](https://www.tutorialspoint.com/c_standard_library/c_function_strcpy.htm)  Copies the string pointed to, by *src* to *dest*. |
| 13 | [**char \*strncpy(char \*dest, const char \*src, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_strncpy.htm)  Copies up to n characters from the string pointed to, by *src* to *dest*. |
| 14 | [**size\_t strcspn(const char \*str1, const char \*str2)**](https://www.tutorialspoint.com/c_standard_library/c_function_strcspn.htm)  Calculates the length of the initial segment of str1 which consists entirely of characters not in str2. |
| 15 | [**char \*strerror(int errnum)**](https://www.tutorialspoint.com/c_standard_library/c_function_strerror.htm)  Searches an internal array for the error number errnum and returns a pointer to an error message string. |
| 16 | [**size\_t strlen(const char \*str)**](https://www.tutorialspoint.com/c_standard_library/c_function_strlen.htm)  Computes the length of the string str up to but not including the terminating null character. |
| 17 | [**char \*strpbrk(const char \*str1, const char \*str2)**](https://www.tutorialspoint.com/c_standard_library/c_function_strpbrk.htm)  Finds the first character in the string *str1* that matches any character specified in *str2*. |
| 18 | [**char \*strrchr(const char \*str, int c)**](https://www.tutorialspoint.com/c_standard_library/c_function_strrchr.htm)  Searches for the last occurrence of the character c (an unsigned char) in the string pointed to by the argument *str*. |
| 19 | [**size\_t strspn(const char \*str1, const char \*str2)**](https://www.tutorialspoint.com/c_standard_library/c_function_strspn.htm)  Calculates the length of the initial segment of *str1* which consists entirely of characters in *str2*. |
| 20 | [**char \*strstr(const char \*haystack, const char \*needle)**](https://www.tutorialspoint.com/c_standard_library/c_function_strstr.htm)  Finds the first occurrence of the entire string *needle* (not including the terminating null character) which appears in the string *haystack*. |
| 21 | [**char \*strtok(char \*str, const char \*delim)**](https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm)  Breaks string *str* into a series of tokens separated by *delim*. |
| 22 | [**size\_t strxfrm(char \*dest, const char \*src, size\_t n)**](https://www.tutorialspoint.com/c_standard_library/c_function_strxfrm.htm)  Transforms the first **n** characters of the string **src** into current locale and places them in the string **dest**. |

Int i;

i = atoi(string);

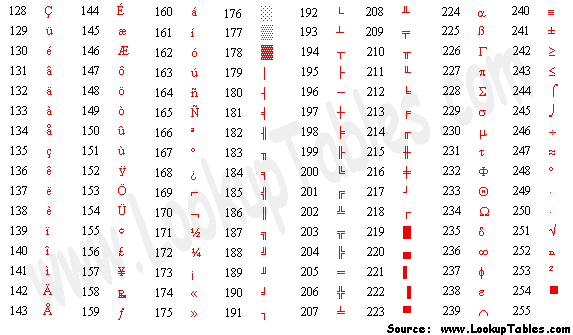
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "header.h"

#include <locale.h>

#include <assert.h>

[int isalnum(int c);](http://www.tutorialspoint.com/ansi_c/c_isalnum.htm)   
The function returns nonzero if c is alphanumeric

[int isalpha(int c);](http://www.tutorialspoint.com/ansi_c/c_isalpha.htm)   
The function returns nonzero if c is alphabetic only

[int iscntrl(int c);](http://www.tutorialspoint.com/ansi_c/c_iscntrl.htm)   
The function returns nonzero if c is a control chracter

[int isdigit(int c);](http://www.tutorialspoint.com/ansi_c/c_isdigit.htm)   
The function returns nonzero if c is a numeric digit

[int isgraph(int c);](http://www.tutorialspoint.com/ansi_c/c_isgraph.htm)   
The function returns nonzero if c is any character for which either isalnum or ispunct returns nonzero.

[int islower(int c);](http://www.tutorialspoint.com/ansi_c/c_islower.htm)   
The function returns nonzero if c is a lower case character.

[int isprint(int c);](http://www.tutorialspoint.com/ansi_c/c_isprint.htm)   
The function returns nonzero if c is space or a character for which isgraph returns nonzero.

[int ispunct(int c);](http://www.tutorialspoint.com/ansi_c/c_ispunct.htm)   
The function returns nonzero if c is punctuation

[int isspace(int c);](http://www.tutorialspoint.com/ansi_c/c_isspace.htm)   
The function returns nonzero if c is space character

[int isupper(int c);](http://www.tutorialspoint.com/ansi_c/c_isupper.htm)   
The function returns nonzero if c is upper case character

[int isxdigit(int c);](http://www.tutorialspoint.com/ansi_c/c_isxdigit.htm)   
The function returns nonzero if c is hexa digit

[int tolower(int c);](http://www.tutorialspoint.com/ansi_c/c_tolower.htm)   
The function returns the corresponding lowercase letter if one exists and if isupper(c); otherwise, it returns c.

[int toupper(int c);](http://www.tutorialspoint.com/ansi_c/c_toupper.htm)   
The function returns the corresponding uppercase letter if one exists and if islower(c); otherwise, it returns c.

#include <math.h>

double sqrt(double x)