C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
<cassert> (assert.h)
<cctype> (ctype.h)
<cerrno> (errno.h)
<cfenv> (fenv.h)
<cfloat> (float.h)
<cinttypes> (inttypes.h)
<ciso646> (iso646.h)
<climits> (limits.h)
<clocale> (locale.h)
<cmath> (math.h)
<csetjmp> (setjmp.h)
<csignal> (signal.h)
<cstdarg> (stdarg.h)
<cstdbool> (stdbool.h)
<cstddef> (stddef.h)
<cstdint> (stdint.h)
<cstdio> (stdio.h)
<cstdlib> (stdlib.h)
<cstring> (string.h)
<ctgmath> (tgmath.h)
<ctime> (time.h)
<cuchar> (uchar.h)
<cwchar> (wchar.h)
<cwctype> (wctype.h)
Containers:
Input/Output:
Multi-threading:
Other:

<cstdlib> (stdlib.h)
functions:
abort
abs
atexit
atof
atoi
atol
atoll
at_quick_exit
bsearch
calloc
div
exit
free
getenv
labs
ldiv
llabs
lldiv
malloc
mblen
mbstowcs
mbtowc
qsort
quick_exit
rand
realloc
srand
strtod
strtof
strtol
strtold
strtoll
strtoul
strtoull
system
wcstombs
wctomb
_Exit
functions (non-standard):
itoa
types:
div_t
ldiv_t
lldiv_t
size_t

function

itoa

<stdlib.h>

```
char *  itoa ( int value, char * str, int base );
```

Convert integer to string (non-standard function)

Converts an integer *value* to a null-terminated string using the specified *base* and stores the result in the array given by *str* parameter.

If *base* is 10 and *value* is negative, the resulting string is preceded with a minus sign (-). With any other *base*, *value* is always considered unsigned.

str should be an array long enough to contain any possible value: `(sizeof(int)*8+1)` for `radix=2`, i.e. 17 bytes in 16-bits platforms and 33 in 32-bits platforms.

Parameters

`value`
Value to be converted to a string.

`str`
Array in memory where to store the resulting null-terminated string.

`base`
Numerical base used to represent the *value* as a string, between `2` and `36`, where `10` means decimal base, `16` hexadecimal, `8` octal, and `2` binary.

Return Value

A pointer to the resulting null-terminated string, same as parameter `str`.

Portability

This function is **not** defined in ANSI-C and is **not** part of C++, but is supported by some compilers.

A standard-compliant alternative for some cases may be sprintf:

- sprintf(str,"%d",value) converts to decimal base.
- sprintf(str,"%x",value) converts to hexadecimal base.
- sprintf(str,"%o",value) converts to octal base.

Example

```
/* itoa example */
#include <stdio.h>
#include <stdlib.h>

int main ()
{
  int i;
  char buffer [33];
  printf ("Enter a number: ");
  scanf ("%d",&i);
  itoa (i,buffer,10);
  printf ("decimal: %s\n",buffer);
  itoa (i,buffer,16);
  printf ("hexadecimal: %s\n",buffer);
  itoa (i,buffer,2);
  printf ("binary: %s\n",buffer);
  return 0;
}
```

Output:

```
Enter a number: 1750
decimal: 1750
hexadecimal: 6d6
binary: 11011010110
```

See also

sprintf	Write formatted data to string (function)
atoi	Convert string to integer (function)
atol	Convert string to long integer (function)

macro constants:
EXIT_FAILURE
EXIT_SUCCESS
MB_CUR_MAX
NULL
RAND_MAX