

<i>lround</i>	<i>Round to Nearest Long Integer (C99)</i>	<i><math.h></i>
	<pre>long int lround(double x);</pre>	
<i>lroundf</i>	<i>long int lroundf(float x);</i>	
<i>lroundl</i>	<i>long int lroundl(long double x);</i>	
<i>Returns</i>	<i>x</i> rounded to the nearest integer, with halfway cases rounded away from zero. If the rounded value is outside the range of the <code>long int</code> type, the result is unspecified and a domain or range error may occur.	
		23.4
<i>malloc</i>	<i>Allocate Memory Block</i>	<i><stdlib.h></i>
	<pre>void *malloc(size_t size);</pre>	
	Allocates a block of memory with <i>size</i> bytes. The block is not cleared.	
<i>Returns</i>	A pointer to the beginning of the block. Returns a null pointer if a block of the requested size can't be allocated.	
		17.2
<i>mblen</i>	<i>Length of Multibyte Character</i>	<i><stdlib.h></i>
	<pre>int mblen(const char *s, size_t n);</pre>	
<i>Returns</i>	If <i>s</i> is a null pointer, returns a nonzero or zero value, depending on whether or not multibyte characters have state-dependent encodings. If <i>s</i> points to a null character, returns zero. Otherwise, returns the number of bytes in the multibyte character pointed to by <i>s</i> ; returns <code>-1</code> if the next <i>n</i> or fewer bytes don't form a valid multibyte character.	
		25.2
<i>mbrlen</i>	<i>Length of Multibyte Character – Restartable (C99)</i>	<i><wchar.h></i>
	<pre>size_t mbrlen(const char * restrict s, size_t n, mbstate_t * restrict ps);</pre>	
	Determines the number of bytes in the array pointed to by <i>s</i> that are required to complete a multibyte character. <i>ps</i> should point to an object of type <code>mbstate_t</code> that contains the current conversion state. A call of <code>mbrlen</code> is equivalent to	
	<code>mbrtowc(NULL, s, n, ps)</code>	
	except that if <i>ps</i> is a null pointer, the address of an internal object is used instead.	
<i>Returns</i>	See <code>mbrtowc</code> .	
		25.5
<i>mbrtowc</i>	<i>Convert Multibyte Character to Wide Character – Restartable (C99)</i>	<i><wchar.h></i>
	<pre>size_t mbrtowc(wchar_t * restrict pwc, const char * restrict s, size_t n, mbstate_t * restrict ps);</pre>	
	If <i>s</i> is a null pointer, a call of <code>mbrtowc</code> is equivalent to	
	<code>mbrtowc(NULL, "", 1, ps)</code>	
	Otherwise, <code>mbrtowc</code> examines up to <i>n</i> bytes in the array pointed to by <i>s</i> to see if	