Description

The C library function **void qsort(void \*base, size\_t nitems, size\_t size, int (\*compar)(const void \*, const void\*))** sorts an array.

Declaration

Following is the declaration for qsort() function.

void qsort(void \*base, size\_t nitems, size\_t size, int (\*compar)(const void \*, const void\*))

Parameters

* **base**-- This is the pointer to the first element of the array to be sorted.
* **nitems**-- This is the number of elements in the array pointed by base.
* **size**-- This is the size in bytes of each element in the array.
* **compar**-- This is the function that compares two elements.

Return Value

This function does not return any value.

Example

The following example shows the usage of qsort() function.

#include <stdio.h>

#include <stdlib.h>

int values[] = { 88, 56, 100, 2, 25 };

int cmpfunc (const void \* a, const void \* b)

{

return ( \*(int\*)a - \*(int\*)b );

}

int main()

{

int n;

printf("Before sorting the list is: \n");

for( n = 0 ; n < 5; n++ ) {

printf("%d ", values[n]);

}

qsort(values, 5, sizeof(int), cmpfunc);

printf("\nAfter sorting the list is: \n");

for( n = 0 ; n < 5; n++ ) {

printf("%d ", values[n]);

}

return(0);

}

Let us compile and run the above program, this will produce the following result:

Before sorting the list is:

88 56 100 2 25

After sorting the list is:

2 25 56 88 100