```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define CAP 1000000000

void input (long int*, long int nr_elements);
void sort (long int*, long int nr_elements);
void output (long int*, long int nr_elements);
void *xcalloc (long int, long int);

int main (int argc, char *argv[])
{
    long int nr_elements;
    long int *arr;

    if (argc != 2)
    {
        fprintf (stderr, "Usage error:%s nr_elements\n",
             argv[0]);
        exit (EXIT_FAILURE);
    }

    nr_elements = atol (argv[1]);
    if (nr_elements <= 0)
    {
        fprintf (stderr, "array dimension must be positive\n");
        exit (EXIT_FAILURE);
    }

    arr = (long int*) xcalloc (nr_elements, sizeof (long int));
    input (arr, nr_elements);
    sort (arr, nr_elements);
    output (arr, nr_elements);

    exit (EXIT_SUCCESS);
}

void input (long int *arr, long int nr_elements)
{
    long int cnt;

    srand (time (0));
    for (cnt=0; cnt < nr_elements; cnt++)
    {
        arr[cnt] = rand () % CAP;
    }
}

void output (long int *arr, long int nr_elements)
{
    long int cnt;

    for (cnt=0; cnt < nr_elements; cnt++)
    {
        printf("arr[%ld]:%ld\n", cnt, arr[cnt]);
    }
}

void *xcalloc (long int nr_elements, long int size_per_element)
{
    void *tmp;

    tmp = calloc (nr_elements, size_per_element);

    if (!tmp)
    {
        fprintf (stderr, "xcalloc:fatal:out of memory\n");
        exit (EXIT_FAILURE);
    }

    return (tmp);
}

```