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

#define CAP 1000000

void *xcalloc(int nr_elements, size_t size_per_element);
void input(int arr[], int nr_elements);
void sort(int arr[], int nr_elements);
void output(int arr[], int nr_elements);

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

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

    nr_elements = atoi(argv[1]);
    if(nr_elements <= 0)
        exit(EXIT_SUCCESS);
    arr = (int*)xcalloc(nr_elements, sizeof(int));
    input(arr, nr_elements);
    sort(arr, nr_elements);
    output(arr, nr_elements);

    exit(EXIT_SUCCESS);
}

void sort(int arr[], int nr_elements){
    int i, j, key;

    for(j=1; j < nr_elements; ++j){

        key = arr[j];
        i = j - 1;

        while(i > -1 && arr[i] > key){
            arr[i+1] = arr[i];
            i = i - 1;
        }

        arr[i+1] = key;
    }
}

void *xcalloc(int nr_elements, size_t size_per_element){
    void *ptr = calloc(nr_elements, sizeof(int));
    if(!ptr){
        fprintf(stderr, "fatal:out of memory\n");
```

```
53          exit(EXIT_FAILURE);
54      }
55
56      return(ptr);
57  }
58
59  void input(int arr[], int nr_elements){
60      int i;
61
62      srand(time(0));
63      for(i=0; i < nr_elements; ++i)
64          arr[i] = rand() % CAP;
65
66      return;
67  }
68
69  void output(int arr[], int nr_elements){
70      int i;
71      for(i=0; i < nr_elements; ++i)
72          printf("arr[%d]:%d\n", i, arr[i]);
73  }
74
```