

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define CAP 1000000
5
6  void *xalloc (int, int);
7  void input (int[], int);
8  void output (int[], int);
9  void sort (int[], int);
10 void merge_sort (int[], int, int);
11
12 int main (int argc, char *argv[])
13 {
14     int *arr;
15     int nr_elements;
16
17     if (argc != 2)
18     {
19         fprintf (stderr, "usage:%s nr_elements\n", argv[0]);
20         exit (EXIT_FAILURE);
21     }
22
23     nr_elements = atoi (argv[1]);
24     arr = xalloc (nr_elements, sizeof (int));
25     input (arr, nr_elements);
26     sort (arr, nr_elements);
27     output (arr, nr_elements);
28
29     return (0);
30 }
31
32 void input (int arr[], int nr_elements)
33 {
34     int cnt;
35
36     for (cnt=0; cnt < nr_elements; cnt++)
37     {
38         arr[cnt] = rand () % CAP;
39     }
40 }
41
42 void output (int arr[], int nr_elements)
43 {
44     int cnt;
45
46     for (cnt=0; cnt < nr_elements; cnt++)
47     {
48         fprintf (stdout, "arr[%d]:%d\n", cnt, arr[cnt]);
49     }
50 }
51
52 void sort (int arr[], int nr_elements)
```

```
53 {  
54     merge_sort (arr, 0, nr_elements-1);  
55 }  
56
```