📄   ⬆   ▶ Run   ⊙ Debug   ■ Stop   ⏄ Share   💾 Save   {} Beautify   ⬇   Language   C   ▾   ⓘ   ⚙

main.c

```c
 8
 9  #include <stdio.h>
10  #include <stdlib.h>
11  #include <time.h>
12  void swap(int *a, int *b) {
13      int temp = *a;
14      *a = *b;
15      *b = temp;
16  }
17  void heap(int arr[], int n, int i) {
18      int largest = i;
19      int left = 2 * i + 1;
20      int right = 2 * i + 2;
21      if (left < n && arr[left] > arr[largest])
22        largest = left;
23      if (right < n && arr[right] > arr[largest])
24        largest = right;
25      if (largest != i) {
26        swap(&arr[i], &arr[largest]);
27        heap(arr, n, largest);
28      }
29  }
30  void heapSort(int arr[], int n) {
31      for (int i = n / 2 - 1; i >= 0; i--)
32        heap(arr, n, i);
33      for (int i = n - 1; i >= 0; i--) {
34        swap(&arr[0], &arr[i]);
```

input

Command line arguments: [_____]

Standard Input: ● Interactive Console   ○ Text

```c
30  void heapSort(int arr[], int n) {
31      for (int i = n / 2 - 1; i >= 0; i--)
32          heap(arr, n, i);
33      for (int i = n - 1; i >= 0; i--) {
34          swap(&arr[0], &arr[i]);
35          heap(arr, i, 0);
36      }
37  }
38  void printArray(int arr[], int n) {
39      for (int i = 0; i < n; ++i)
40          printf("%d ", arr[i]);
41      printf("\n");
42  }
43  int main() {
44      int n, i, j, temp,c;
45      int arr[10];
46      printf("Enter the number of elements to be sorted: \n");
47      scanf("%d",&n);
48      printf("Genrating random nos.:\n");
49      srand(time(0));
50      for (c = 0; c < n; c++){
51      arr[c]=rand()%1000 +1;
52      printf("%d\t\n",arr[c]);
53      }
54      clock_t begin = clock();
55      heapSort(arr, n);
56      printf("Sorted array is \n");
57      printArray(arr, n);
58      clock_t end = clock();
59      double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
60      printf("\n\nEXECUTION TIME : %.10fseconds\n", time_spent);
61  }
```
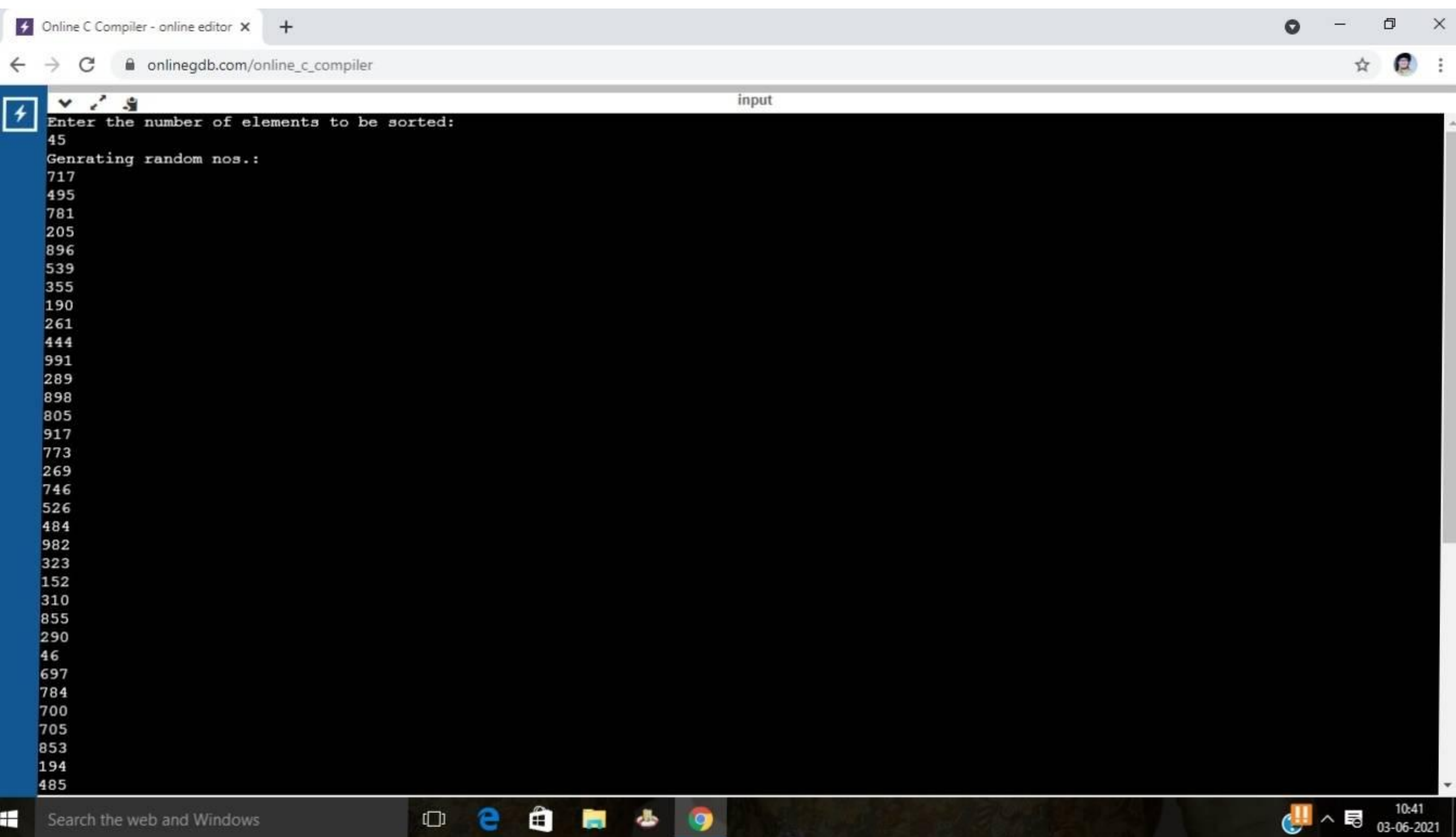
```
Enter the number of elements to be sorted:
45
Genrating random nos.:
717
495
781
205
896
539
355
190
261
444
991
289
898
805
917
773
269
746
526
484
982
323
152
310
855
290
46
697
784
700
705
853
194
485
```

```
526
484
982
323
152
310
855
290
46
697
784
700
705
853
194
485
409
441
375
763
630
636
207
972
924
104
777
Sorted array is
46 104 152 190 194 205 207 261 269 289 290 310 323 355 375 409 441 444 484 485 495 526 539 630 636 697 700 705 717 746 763 773 777 781 784 805
853 855 896 898 917 924 972 982 991


EXECUTION TIME : 0.0000220000seconds


...Program finished with exit code 0
Press ENTER to exit console.
```