**NON-RECURSIVE SELECTION SORT:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

clock\_t start,end;

double time;

int a[100000], n, i, j, loc, temp;

printf("Enter number of elements in the array:\n");

scanf("%d", &n);

printf("The elements in the array are: \n");

for (i = 0; i < n; i++)

{

a[i] = rand()%100;

printf("%d\n",a[i]);

}

start = clock();

for(i = 0; i < n - 1; i++)

{

loc=i;

for(j = i + 1; j < n; j++)

{

if(a[loc] > a[j])

loc=j;

}

if(loc != i)

{

temp=a[i];

a[i]=a[loc];

a[loc]=temp;

}

}

end = clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("Time taken : %lf\n",time);

printf("Sorted Array using Selection Sort is:\n");

for(i = 0; i < n; i++)

printf("%d\n", a[i]);

return 0;

}

**RECURSIVE SELECTION SORT:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void Sel\_Sort(int [], int, int);

int main()

{

clock\_t start,end;

double time;

int arr[100000], n, i;

printf("Enter the size of the array:\n");

scanf("%d", &n);

printf("Enter the elements in the array:\n");

for (i = 0; i < n; i++)

{

arr[i] = rand()%100;

printf("%d ",arr[i]);

}

start = clock();

Sel\_Sort(arr, 0, n);

end = clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken : %lf\n",time);

printf("\nThe sorted list in ascending order is\n");

for (i = 0; i < n; i++)

{

printf("%d ", arr[i]);

}

return 0;

}

void Sel\_Sort(int arr[], int i, int n)

{

int j;

int min = i;

for (j = i + 1; j < n; j++)

{

if (arr[j] < arr[min]) {

min = j;

}

}

int temp = arr[min];

arr[min] = arr[i];

arr[i] = temp;

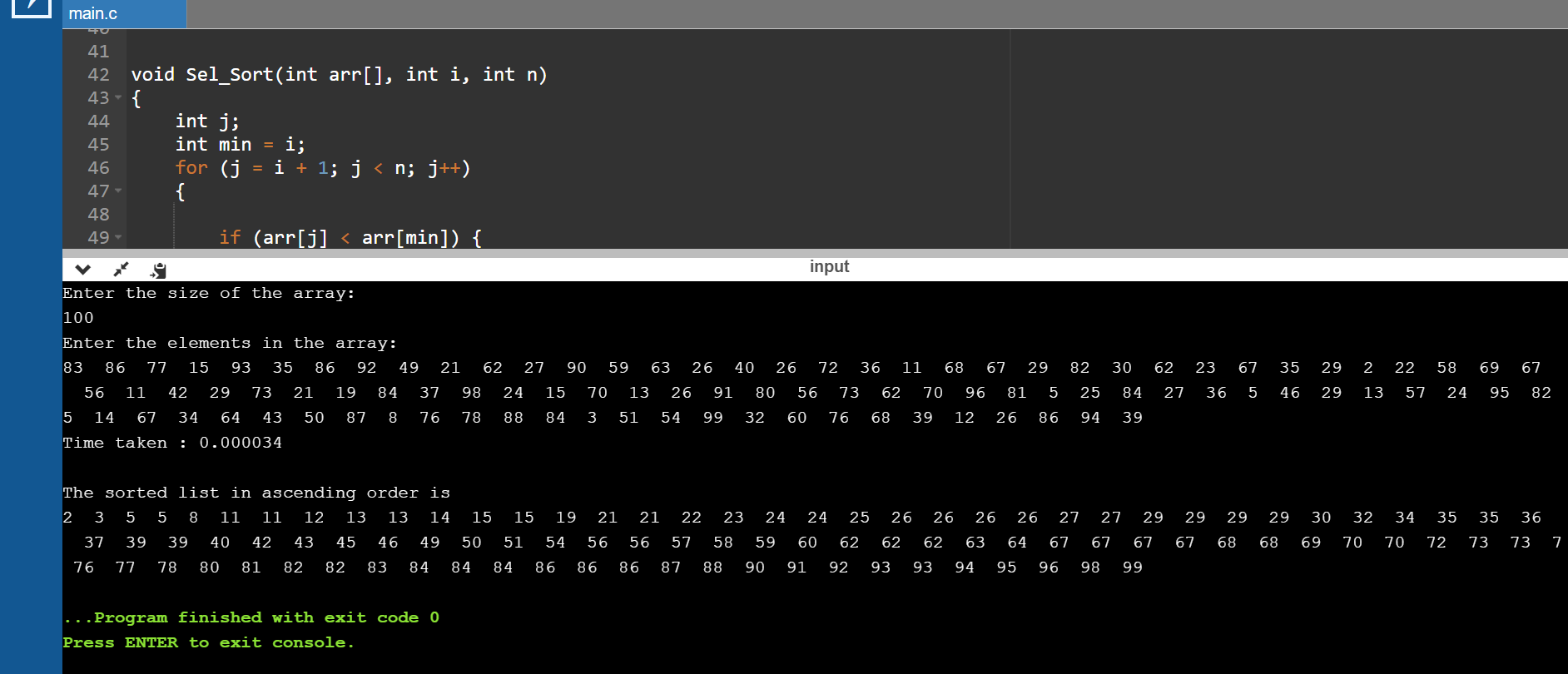
if (i + 1 < n) {

Sel\_Sort(arr, i + 1, n);

}

}

OUTPUT:



|  |  |
| --- | --- |
| **N** | **Time Taken** |
| 50 | 0.000015 |
| 100 | 0.000034 |
| 200 | 0.000103 |
| 500 | 0.000479 |
| 1000 | 0.001893 |
| 5000 | 0.041123 |
| 10000 | 0.156707 |
| 15000 | 0.405502 |
| 20000 | 0.607911 |
| 25000 | 1.080052 |

**RECURSIVE BUBBLE SORT:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void bubbleSort(int [], int);

int main()

{

clock\_t start,end;

double time;

int arr[100000], n, i;

printf("Enter the size of the array:\n");

scanf("%d", &n);

printf("Enter the elements in the array:\n");

for (i = 0; i < n; i++)

{

arr[i] = rand()%100;

printf("%d ",arr[i]);

}

start = clock();

bubbleSort(arr, n);

end = clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken : %lf\n",time);

printf("\nThe sorted list in ascending order is\n");

for (i = 0; i < n; i++)

{

printf("%d ", arr[i]);

}

return 0;

}

void bubbleSort(int arr[], int n)

{

for (int i = 0; i < n - 1; i++)

{

if (arr[i] > arr[i + 1]) {

int temp = arr[i];

arr[i] = arr[i+1];

arr[i+1] = temp;

}

}

if (n - 1 > 1) {

bubbleSort(arr, n - 1);

}

}

OUTPUT:

|  |  |
| --- | --- |
| **N** | **Time Taken** |
| 50 | 0.000016 |
| 100 | 0.000044 |
| 200 | 0.000166 |
| 500 | 0.000721 |
| 1000 | 0.002964 |
| 5000 | 0.068924 |
| 10000 | 0.328939 |
| 15000 | 0.992609 |
| 20000 | 1.382367 |
| 25000 | 2.260302 |