**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. IIIrd Sem**

**Subject Code:** CS211

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

/\*

Q. Lab Question: Compare the performance of Selection Sort, Bubble Sort and Insertion Sort. Input: Ascending Order Sorted, Descending Order Sorted and Random Integer. Output: Time measurement in Seconds. Document: Plot a comparison chart

\*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void bubblesort (int \*arr, int size);

void selectionsort (int \*arr, int size);

void insertionsort (int \*arr, int size);

int main()

{

int \*arr, \*temparr, size, i, j;

clock\_t start, end;

double bubble\_t, insertion\_t, selection\_t;

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

scanf ("%d", &size);

arr = (int \*) malloc (size \* sizeof(int));

temparr = (int \*) malloc (size \* sizeof(int));

for (j=0; j<3;++j)

{

if (j==0)

{

printf ("\n\n\n###########################\n");

printf ("ASCENDING ORDER INPUT GIVEN");

printf ("\n###########################\n");

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

{

arr[i] = i+1;

}

}

else if (j==1)

{

printf ("\n\n\n############################\n");

printf ("DESCENDING ORDER INPUT GIVEN");

printf ("\n############################\n");

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

{

arr[i] = size - i;

temparr[i] = arr[i];

}

}

else

{

printf ("\n\n\n##############################\n");

printf ("RANDOMLY GENERATED INPUT GIVEN");

printf ("\n##############################\n");

srand (time(0));

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

{

arr [i] = rand() % 100;

temparr [i] = arr[i];

}

}

start = clock ();

bubblesort (arr, size);

end = clock ();

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

printf ("\nTime taken by bubble sort is %lf.", bubble\_t);

if (j==2 || j==1)

{

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

{

arr[i] = temparr [i];

}

}

start = clock ();

selectionsort (arr, size);

end = clock ();

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

printf ("\nTime taken by selection sort is %lf.", selection\_t);

if (j==2 || j==1)

{

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

{

arr[i] = temparr [i];

}

}

start = clock();

insertionsort (arr, size);

end = clock ();

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

printf ("\nTime taken by insertion sort is %lf.", insertion\_t);

}

printf ("\n\n");

return 0;

}

void bubblesort (int \*arr, int size)

{

int i, j, temp;

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

{

for (j = 0; j<size-i-1; ++j)

{

if (arr[j] > arr[j+1])

{

temp = arr [j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

void selectionsort (int \*arr, int size)

{

int i, j, temp, ind;

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

{

ind = i;

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

{

if (arr[j] < arr[ind])

ind = j;

}

temp = arr [i];

arr [i] = arr[ind];

arr[ind] = temp;

}

}

void insertionsort (int \*arr, int size)

{

int i, j, key;

for (i=1; i<size; ++i)

{

key = arr[i];

for (j=i-1; (j>=0 && arr[j]>key); --j)

{

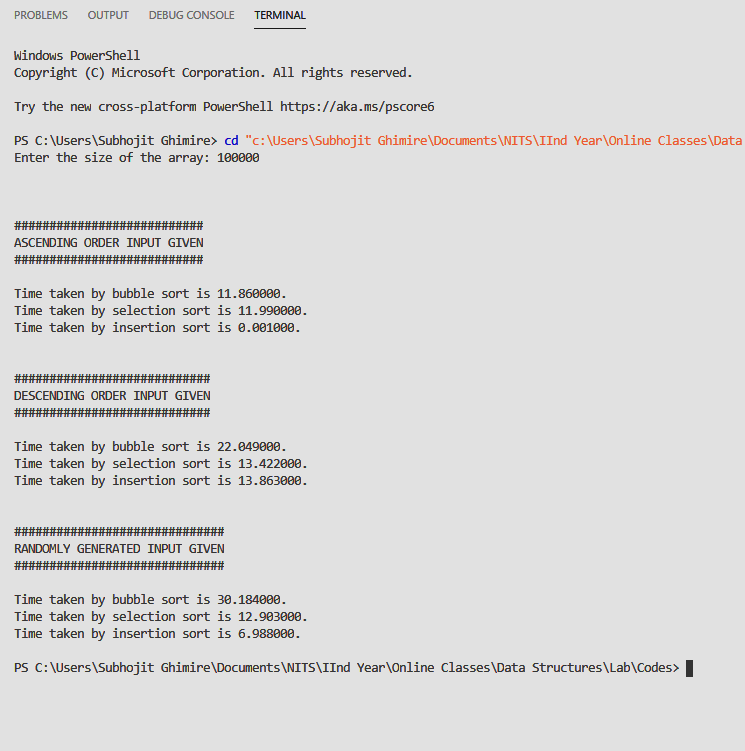
arr[j+1] = arr[j];

}

arr [j+1] = key;

}

}

OUTPUT

GRAPHICAL COMPARISON REPRESENTATION