**Computer Science and Engineering Discipline**

**Khulna University, Khulna - 9208**

Roll#:

Name:Tanvir Hassan

email:tanvir1908@cseku.ac.bd

Mobile:01989567104

**Assignment on Insertion Sorting**

Implement the insertion sorting algorithm **using C/C++** (Any other programming language **is not allowed**). Input will be the different numbers of positive integers. Execute your program and fill in the following Table I and provide other information.

1. **Write the code of your program below.**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int n,i,temp,j,data\_movement = 0,comparisn = 0;

printf("Enter how many data do you want : ");

scanf("%d",&n);

int A[n];

printf("Data : \n");

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

{

A[i] = 1+rand();

}

printf("Before sorting : ");

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

{

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

}

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

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

{

comparisn++;

j=i;

while (j >0&A[j-1]>A[j])

{

data\_movement++;

temp=A[j];

A[j]=A[j-1];

A[j-1]=temp;

j--;

}

}

printf("Sorted list in ascending order:\n");

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

{

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

}

printf("\n\nTotal comparisn : %d\n",comparisn);

printf("\nTotal data movement : %d\n",data\_movement);

return 0 ;

}

Table I: Simulation results of Insertion sorting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No. of Integers** | **No. of data Comparisons** | **No. of data movement** | **Execution time** |  |
| 100 | 99 | 2360 | 2.089 s |  |
| 500 | 499 | 62420 | 2.852 s |  |
| 1000 | 999 | 248145 | 3.668 s |  |
| 10000 | 9999 | 24952888 | 3.843 s |  |
| 1000000 | invalid | invalid | invalid |  |
| 10000000 | invalid | invalid | invalid |  |

**I. Input and Output of the program**

1. Input of 100 integers:

42 18468 6335 26501 19170 15725 11479 29359 26963 24465 5706 28146 23282 16828 9962 492 2996 11943 4828 5437 32392 14605 3903 154 293 12383 17422 18717 19719 19896 5448 21727 14772 11539 1870 19913 25668 26300 17036 9895 28704 23812 31323 30334 17674 4665 15142 7712 28254 6869 25548 27645 32663 32758 20038 12860 8724 9742 27530 779 12317 3036 22191 1843 289 30107 9041 8943 19265 22649 27447 23806 15891 6730 24371 15351 15007 31102 24394 3549 19630 12624 24085 19955 18757 11841 4967 7377 13932 26309 16945 32440 24627 11324 5538 21539 16119 2083 22930 16542

Output of 100 integers:

42 154 289 293 492 779 1843 1870 2083 2996 3036 3549 3903 4665 4828 4967 5437 5448 5538 5706 6335 6730 6869 7377 7712 8724 8943 9041 9742 9895 9962 11324 11479 11539 11841 11943 12317 12383 12624 12860 13932 14605 14772 15007 15142 15351 15725 15891 16119 16542 16828 16945 17036 17422 17674 18468 18717 18757 19170 19265 19630 19719 19896 19913 19955 20038 21539 21727 22191 22649 22930 23282 23806 23812 24085 24371 24394 24465 24627 25548 25668 26300 26309 26501 26963 27447 27530 27645 28146 28254 28704 29359 30107 30334 31102 31323 32392 32440 32663 32758

2. Input of **the last** 100 integers out of 10000000 : invalid.

Output of **the last** 100 integers out of 10000000 : invalid.