**Program No. 15(b)**

Objective:-WAP related to Sorting

#include <stdio.h>

void Aakash\_HeapSort()

{

int heap[50], no, i, j, c, root, temp;

printf("\nEnter size of Array or List:");

scanf("%d", &no);

printf("\nEnter the element in Array:\n");

for (i = 0; i < no; i++)

scanf("%d", &heap[i]);

for (i = 1; i < no; i++)

{

c = i;

do

{

root = (c - 1) / 2;

if (heap[root] < heap[c])

{

temp = heap[root];

heap[root] = heap[c];

heap[c] = temp;

}

c = root;

} while (c != 0);

}

for (j = no - 1; j >= 0; j--)

{

temp = heap[0];

heap[0] = heap[j];

heap[j] = temp;

root = 0;

do

{

c = 2 \* root + 1;

if ((heap[c] < heap[c + 1]) && c < j-1)

c++;

if (heap[root]<heap[c] && c<j)

{

temp = heap[root];

heap[root] = heap[c];

heap[c] = temp;

}

root = c;

} while (c < j);

}

printf("\n The sorted array is : ");

for (i = 0; i < no; i++)

printf("\t %d", heap[i]);

}

void Aakash\_quicksort(int a[], int min, int max)

{

int key, i, j, temp;

if (min < max)

{

key = min;

i = min;

j = max;

while (i < j)

{

while (a[i]<=a[key] && i<=max)

{

i++;

}

while (a[j]>a[key] && j>=min)

{

j--;

}

if (i < j)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

temp = a[j];

a[j] = a[key];

a[key] = temp;

quicksort(a, min, j - 1);

quicksort(a, j + 1, max);

}

}

int main()

{

int n,a[50],size,i;

printf("Enter the value between (1-3)=\n");

printf("\n\nFor Quick Sort Press 1:");

printf("\n\nFor Heap Sort Press 2:\n");

scanf("%d",&n);

printf("\n");

switch(n)

{

case 1:

{

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

scanf("%d", &size);

printf("Enter the elements to be sorted:\n");

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

{

scanf("%d", &a[i]);

}

Aakash\_quicksort(a, 0, size - 1);

printf("After applying quick sort\n");

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

{

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

}

printf("\n");

return 0;

}

case 2:

{

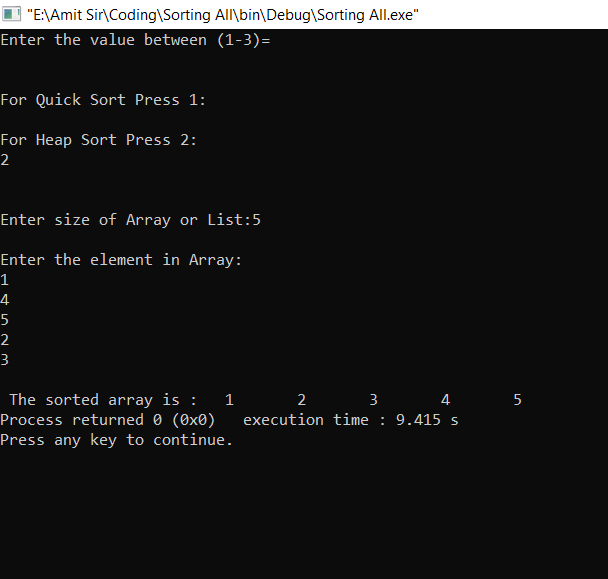
Aakash\_HeapSort();

}

}

}

Output:-



Operation performed on Sorting