WEEK 10

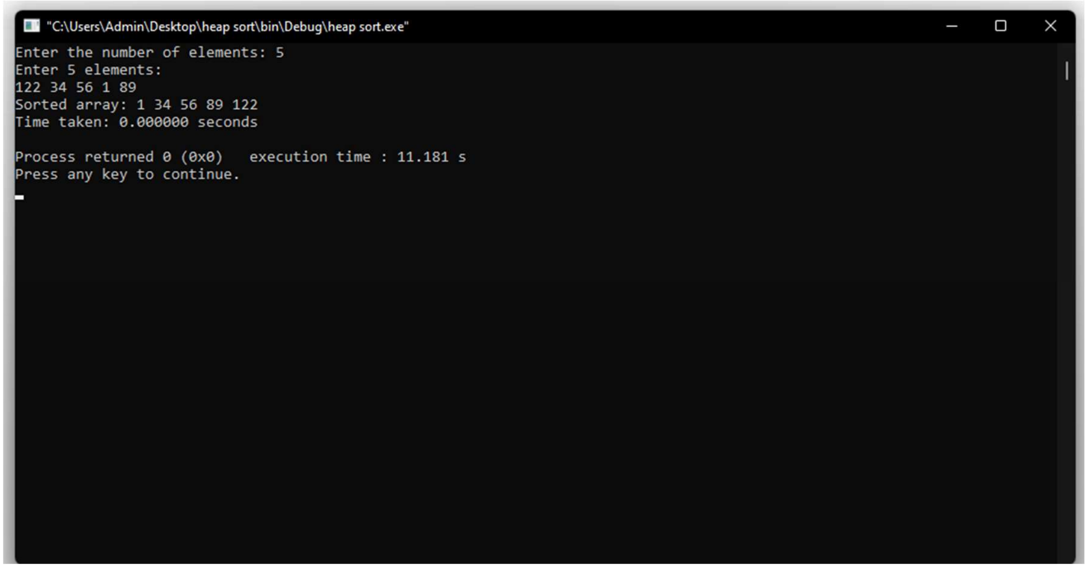
Q. Sort a given set of N integer elements using Heap Sort technique and compute its time taken.

CODE:

```
#include <stdio.h>
#include <time.h>

void heapify(int arr[], int n, int i) {
int largest = i;     int left = 2 * i +
1;     int right = 2 * i + 2;

   if (left < n && arr[left] > arr[largest])
largest = left;

   if (right < n && arr[right] > arr[largest])
largest = right;

   if (largest != i) {        int
temp = arr[i];        arr[i] =
arr[largest];
arr[largest] = temp;
heapify(arr, n, largest);
   }
}
```

```
void heapSort(int arr[], int n) {
for (int i = n / 2 - 1; i >= 0; i--)
heapify(arr, n, i);

   for (int i = n - 1; i > 0; i--) {
int temp = arr[0];
arr[0] = arr[i];        arr[i] =
temp;        heapify(arr, i, 0);
   }
}

int main() {
int n;
   printf("Enter the number of elements: ");
scanf("%d", &n);

   int arr[n];
   printf("Enter %d elements:\n", n);
for (int i = 0; i < n; i++)
scanf("%d", &arr[i]);

   clock_t start_time = clock();
heapSort(arr, n);     clock_t
end_time = clock();
printf("Sorted array: ");     for
(int i = 0; i < n; i++)
printf("%d ", arr[i]);
printf("\n");

    double time_taken = (double)(end_time - start_time) / CLOCKS_PER_SEC;
printf("Time taken: %f seconds\n", time_taken);

    return 0;
}
```

OUTPUT:



```
Enter the number of elements: 5
Enter 5 elements:
122 34 56 1 89
Sorted array: 1 34 56 89 122
Time taken: 0.000000 seconds

Process returned 0 (0x0)   execution time : 11.181 s
Press any key to continue.
```