```cpp
#include<iostream>
#include<stdlib.h>
#include<time.h>
#include<chrono>
#include<ctime>

void build_MaxHeap(int[],int,int);
void max_heapify(int[],int,int);
void selection_sort(int [],int);

using namespace std;
int main()
{   //variables, random seed, input, dynamic array
    int n,num;
    srand(time(NULL));
    cout<<"Enter array size"<<endl;
    cin>>n;
    int *arr= new int[n];
    //final time
    double l1;

    //int arr[]={30,25,1,56,33};

    //generate n random numbers
    for(int x=0; x<n;x++)
        {
        num = rand() %200+(-100);
        arr[x]=num;
        }
    //displaying the original array
    cout<<"Original"<<endl;
    for(int x=0; x<n; x++)
        cout<<arr[x]<<" ";

    //call max_heap
    build_MaxHeap(arr,0,n);
    cout<<"\n"<<"""After Build Heapify"<<endl;
    for(int x=0; x<n; x++)
        cout<<arr[x]<<" ";


    for(int x=0; x<=100;x++)
    {
    //generate 10000 numbers
    for(int x=0; x<n;x++)
    {
        int num = rand() %200+(-100);
        arr[x]=num;
    }
    //start clock, sort
        auto start = chrono::high_resolution_clock::now();
        for(int x=n; x>0; x--)
            {
                swap(arr[0],arr[x-1]);
                max_heapify(arr,0,x);
            }

    //end clock, increment duration
        auto end = chrono::high_resolution_clock::now();
   //increment duration
         l1 += chrono::duration_cast<chrono::microseconds>(end - start).count();
    }
```

```
    cout<<"\n Duration: "<< l1 <<" microseconds"<< endl;

   //array for selection sort
   int selArray[10];
   for(int i =0; i<10;i++)
   {
           num=rand() %200+(-100);
           selArray[i]=num;
   }
   // for(int i =0; i<10; i++)
//         cout<<selArray[i]<<" ";
  cout<<endl;
  selection_sort(selArray,10);
  cout<<endl;
   for(int i =0; i<10; i++)
           cout<<selArray[i]<<" ";

  //random 10 array
   int maxH[10];
   for(int i =0; i<10;i++)
   {
           num=rand() %200+(-100);
           maxH[i]=num;
   }
   //print original
   cout<<"Original Array: "<<endl;
   for(int i=0 ;i<10;i++)
           cout<<maxH[i]<<" ";
   cout<<endl;
   build_MaxHeap(maxH,0,10);
   for(int i=9; i>0;i--)
   {
           swap(maxH[0],maxH[i]);
           max_heapify(maxH,0,i);
   }
   cout<<"After Heap Sort: "<<endl;
   for(int i=0; i<10; i++)
           cout<<maxH[i]<<" ";


 delete[] arr;
 delete arr;
}

void build_MaxHeap(int arr[],int start, int end)
{
    int size=end-start;
    for(int x=end-1; x>=start; x--)
    {
        max_heapify(arr,x,size);
    }
}

void max_heapify(int arr[],int i, int size)
{
    int max = i;
    int left=2*i+1;
    int right = 2*i+2;
    if(left<size && arr[left]>arr[max])
        max=left;
    if(right<size && arr[right]>arr[max])
        max=right;
```

```
    if(max!= i)
    {
        swap(arr[max],arr[i]);
        max_heapify(arr,max,size);
    }
}

void selection_sort(int arr[],int end)
{       int smallest;
        for(int i =0; i<end; i++)
        {       smallest=i;
                for(int j=i; j<end; j++)
                {

                        if(arr[j]<arr[smallest])
                                smallest=j;

                }
                swap(arr[i],arr[smallest]);

        }

}
```