

```

#include <stdio.h>
int main()
{
    int n, i, j, temp;
    int arr[64];

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    for (i = 1 ; i <= n - 1; i++)
    {
        j = i;
        while ( j > 0 && arr[j-1] > arr[j])
        {
            temp    = arr[j];
            arr[j]  = arr[j-1];
            arr[j-1] = temp;
            j--;
        }
    }
    printf("Sorted list in ascending order:\n");
    for (i = 0; i <= n - 1; i++)
    {
        printf("%d\n", arr[i]);
    }
    return 0;
}

```

```

#include <stdio.h>
int main()
{
    int a[100], n, i, j, position, swap;
    printf("Enter number of elementsn");
    scanf("%d", &n);
    printf("Enter %d Numbersn", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    for(i = 0; i < n - 1; i++)
    {

```

```

position=i;
for(j = i + 1; j < n; j++)
{
    if(a[position] > a[j])
        position=j;
}
if(position != i)
{
    swap=a[i];
    a[i]=a[position];
    a[position]=swap;
}
}
printf("Sorted Array:n");
for(i = 0; i < n; i++)
    printf("%dn", a[i]);
return 0;
}

```

[8:32 PM, 5/6/2020] Naveen: selection sort

[8:32 PM, 5/6/2020] Naveen: bubble sort:.....// C program for implementation of Bubble sort

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

```

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

```

// A function to implement bubble sort

```
void bubbleSort(int arr[], int n)
```

```

{
    int i, j;
    for (i = 0; i < n-1; i++)

        // Last i elements are already in place
        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}

```

```
/* Function to print an array */
```

```

void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

```

// Driver program to test above functions

```

int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
*/

```

#include <stdio.h>

void main()

```

{
    int heap[10], no, i, j, c, root, temp;

    printf("\n Enter no of elements :");
    scanf("%d", &no);
    printf("\n Enter the nos : ");
    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]) /* to create MAX heap array */
            {
                temp = heap[root];
                heap[root] = heap[c];
                heap[c] = temp;
            }
            c = root;
        }
    }
}

```

```

    } while (c != 0);
}

printf("Heap array : ");
for (i = 0; i < no; i++)
    printf("%d\t", heap[i]);
for (j = no - 1; j >= 0; j--)
{
    temp = heap[0];
    heap[0] = heap[j] /* swap max element with rightmost leaf element */
    heap[j] = temp;
    root = 0;
    do
    {
        c = 2 * root + 1; /* left node of root element */
        if ((heap[c] < heap[c + 1]) && c < j-1)
            c++;
        if (heap[root]<heap[c] && c<j) /* again rearrange to max heap array */
        {
            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]);
printf("\n Complexity : \n Best case = Avg case = Worst case = O(n logn) \n");
}

```