```c
// C program to convert min Heap to max Heap

#include <stdio.h>

void swap(int* a, int* b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

// to heapify a subtree with root at given index
void MinHeapify(int arr[], int i, int N)
{
    int l = 2 * i + 1;
    int r = 2 * i + 2;
    int smallest = i;

    if (l < N && arr[l] < arr[i])
        smallest = l;
    if (r < N && arr[r] < arr[smallest])
        smallest = r;
    if (smallest != i) {
        swap(&arr[i], &arr[smallest]);
        MinHeapify(arr, smallest, N);
    }
}

// This function basically builds max heap
void convertMinHeap(int arr[], int N)
{
    // Start from bottommost and rightmost
    // internal mode and heapify all internal
    // modes in bottom up way
    for (int i = (N - 2) / 2; i >= 0; --i)
        MinHeapify(arr, i, N);
}

// A utility function to print a given array
```

```
40: // of given size
41: void printArray(int* arr, int size)
42: {
43:     for (int i = 0; i < size; ++i)
44:         printf("%d ", arr[i]);
45: }
46:
47: // Driver's code
48: int main()
49: {
50:     // array representing Min Heap
51:     int arr[] = { 20,18,10,12,9,9,3,5,6,8 };
52:     int N = sizeof(arr) / sizeof(arr[0]);
53:
54:     printf("Max Heap array : ");
55:     printArray(arr, N);
56:
57:     // Function call
58:     convertMinHeap(arr, N);
59:
60:     printf("\nMin Heap array : ");
61:     printArray(arr, N);
62:
63:     return 0;
64: }
65:
```