

```

1: // Max-Heap data structure in C
2:
3: #include <stdio.h>
4:
5: int size = 0;
6: void swap(int *a, int *b)
7: {
8:     int temp = *b;
9:     *b = *a;
10:    *a = temp;
11: }
12: void heapify(int array[], int size, int i)
13: {
14:     if (size == 1)
15:     {
16:         printf("Single element in the heap");
17:     }
18:     else
19:     {
20:         int smallest = i;
21:         int l = 2 * i + 1;
22:         int r = 2 * i + 2;
23:         if (l < size && array[l] < array[smallest])
24:             smallest = l;
25:         if (r < size && array[r] < array[smallest])
26:             smallest = r;
27:         if (smallest != i)
28:         {
29:             swap(&array[i], &array[smallest]);
30:             heapify(array, size, smallest);
31:         }
32:     }
33: }
34: void insert(int array[], int newNum)
35: {
36:     if (size == 0)
37:     {
38:         array[0] = newNum;
39:         size += 1;

```

```

40:     }
41:     else
42:     {
43:         array[size] = newNum;
44:         size += 1;
45:         for (int i = size / 2 - 1; i >= 0; i--)
46:         {
47:             heapify(array, size, i);
48:         }
49:     }
50: }
51: void deleteRoot(int array[], int num)
52: {
53:     int i;
54:     for (i = 0; i < size; i++)
55:     {
56:         if (num == array[i])
57:             break;
58:     }
59:
60:     swap(&array[i], &array[size - 1]);
61:     size -= 1;
62:     for (int i = size / 2 - 1; i >= 0; i--)
63:     {
64:         heapify(array, size, i);
65:     }
66: }
67: void printArray(int array[], int size)
68: {
69:     for (int i = 0; i < size; ++i)
70:         printf("%d ", array[i]);
71:     printf("\n");
72: }
73: int main()
74: {
75:     int array[10];
76:
77:     insert(array, 3);
78:     insert(array, 4);

```

```
79:  insert(array, 9);
80:  insert(array, 5);
81:  insert(array, 2);
82:
83:  printf("Min-Heap array: ");
84:  printArray(array, size);
85:
86:  deleteRoot(array, 4);
87:
88:  printf("After deleting an element: ");
89:
90:  printArray(array, size);
91: }
```