Name: Mohammad Danish Eqbal Branch: CSE Section: A'
Roll No.: 62 Reg. No.: 19030513

- Answers -

Code:

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

int H[50];
int size = -1;

int parent(int i)
{ return (i-1)/2; }

void up_shift(int i)
{ while (i>0 && H[parent(i)]< H[i])
      { int tmp = H[i];
        H[i] = H[parent(i)];
        H[parent(i)] = tmp;
        i = parent(i);
      }
}

int lchild(int i)
{ return ((2*i)+1); }

int rchild(int i)
{ return ((2*i)+2); }
```

```
void downshift (int i)
{ int maxIndex = i;
  int l = lchild(i);
  if (i<=size && H[l] > H[maxIndex])
    { maxIndex = l; }
  int r = rchild (i);
  if (r<=size && H[r] > H[maxIndex])
    { maxIndex = r; }
  if (i != maxIndex)
  { int temp = H[i];
    H[i] = H[maxIndex];
    H[maxIndex] = temp;
    downshift (maxIndex);
  }
}

void insert (int p)
  { size = size+1;
    H[size] = p;
    upshift (size);
  }

int max-extract() {
  int result = H[0];
  H[0] = H[size];
  size = size-1;
  downshift (0);
  return result;
  }
```

```
void priority_change(int i, int p)
 { int oldp = H[i];
   H[i] = p;
  if (p > oldp)
  { up_shift(i);}
  else
  {downshift(i);}
 }

int getMax(){
            return H[0];
           }

void priority Remove(int i) {
       H[i] = getMax() +1;
       up_shift(i);
       max_extract();
      }

int main()
{
 int N=0;
 int val, rem;
 int i=0, j=0, k=0, l=0;
 printf("Enter number of element to insert:\n");
 scanf("%d",&N);
 while(N!=0)
 { scanf("%d" &val);
   insert(val);
   N--;
 }
```

```
printf ("Priority Queue :");

while (i <= size)
 { printf ("%d", H[i]);
   i++;
 }
printf ("\n");
printf ("Max^m priority node : %d", max_extract ());
printf ("Extracting max^m: ");
while (j ≤ size)
   { printf ("%d\t", H[j]);
     j++;
   }
printf ("\n");
priority change (100, 12),
printf ("Priority queue after priority change :");
while (k <= size)
      { printf ("%d\t", H[k]);
        k++
      }
printf ("\n");
printf ("Enter priority to remove : ");
scanf ("%d", &rem);
priorityRemove (rem);
printf (" Priority Queue after removing the element : \n");

while (l <= size)
  { printf ("%d \t", H [l]);
    l++;
  }
return 0;
}
```

Sample Input/Output After Execution:

Enter number of element to insert:
3

5 4 6
Priority Queue: 6 4 5
Maximum Priority Node: 6 Extracting Maximum: 5 4
Priority Queue after priority change: 12 4
Enter priority to remove: 4
Priority Queue after removing element: 12