Test Cases for Question 1

Part 1 - Test cases without repetitive elements

1_One element

INPUT

```
1
5
```

OUTPUT

```
5
0
```

2_Two elements

INPUT

```
2
78 76
```

OUTPUT

```
76 78
1
```

3_Sorted in ascending order

INPUT

```
20
-999 -875 -500 -387 -252 -222 -111 -98 0 12 212 312 412 512 612 712 891
910 918 999
```

OUTPUT

```
-999 -875 -500 -387 -252 -222 -111 -98 0 12 212 312 412 512 612 712 891
910 918 999
121
```

4_Sorted in descending order

INPUT

```
20
999 875 500 387 252 222 111 98 0 -12 -212 -312 -412 -512 -612 -712 -891 -
910 -918 -999
```

OUTPUT

```
-999 -918 -910 -891 -712 -612 -512 -412 -312 -212 -12 0 98 111 222 252
387 500 875 999
105
```

5_Random sequence_1

INPUT

```
24
1 11 111 2 22 222 111 121 131 125 145 165 178 201 245 224 236 241 5 7 10
201 241 250
```

OUTPUT

```
1 2 5 7 10 11 22 111 111 121 125 131 145 165 178 201 201 222 224 236 241
241 245 250
153
```

6_Random sequence_2

INPUT

```
21
15 7 22 17 25 24 33 28 44 42 99 44 100 99 412 289 534 432 675 634 765
```

OUTPUT

```
7 15 17 22 24 25 28 33 42 44 44 99 99 100 289 412 432 534 634 675 765
129
```

7_Front elements are sorted in ascending order

INPUT

```
21
10 20 30 40  50 170 190 110 70 160 90 210 120 60 180 130 100 200 80 140
150
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
126
```

8_Front elements are sorted in descending order

INPUT

```
21
50 40 30 20 10 170 190 110 70 160 90 210 120 60 180 130 100 200 80 140
150
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
127
```

9_Middle elements are sorted in ascending order

INPUT

```
21
140 50 80 20 60 200 150 190 90 100 110 120 130 10 180 70 170 40 210 160
30
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
130
```

10_Middle elements are sorted in descending order

INPUT

```
21
140 50 80 20 60 200 150 190 130 120 110 100 90 10 180 70 170 40 210 160
30
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
126
```

11_Back elements are sorted in ascending order

INPUT

```
21
130 80 100  40 140 50 110 20 90 70 10 160 150  30 120 60 170 180 190 200
210
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
128
```

12_Back elements are sorted in descending order

INPUT

```
21
130 80 100 40 140 50 110 20 90 70 10 160 150 30 120 60 210 200 190 180
170
```

OUTPUT

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
210
120
```

Part 2 - Test cases with repetitive elements

13_Same elements

INPUT

```
25
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
12
```

OUTPUT

```
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
12
69
```

14_Repitition of smallest element

INPUT

```
22
10 20 210 190 150 100 160 70 80 60 220 10 10 200 140 120 170 180 90 40 10
10
```

OUTPUT

```
10 10 10 10 10 20 40 60 70 80 90 100 120 140 150 160 170 180 190 200 210
220
129
```

15_Repitition of largest element

INPUT

```
22
220 220 210 190 150 100 160 70 80 60 220 220 10 200 140 120 170 180 90 40
110 220
```

OUTPUT

```
10 40 60 70 80 90 100 110 120 140 150 160 170 180 190 200 210 220 220 220
220 220
121
```

16_Repitition of some random element

INPUT

```
22
130 20 110 190 150 100 160 70 80 60 220 110 10 200 140 120 170 180 90 110
110 50
```

OUTPUT

```
10 20 50 60 70 80 90 100 110 110 110 110 120 130 140 150 160 170 180 190
200 220
132
```

Test Cases for Question 2

1_Elements are inserted in ascending order

INPUT

```
i 11
i 22
i 33
m
i 44
i 77
m
i 88
i 111
e
e
e
e
e
e
e
s
```

OUTPUT

```
11
11
11
22
33
44
77
88
111
```

2_ Elements are inserted in descending order

INPUT

```
i 101
i 80
i 75
m
i 44
i 36
m
i 10
i 2
e
e
e
e
e
e
e
```

```
s
```

OUTPUT

```
75
36
2
10
36
44
75
80
101
```

3_ Elements are inserted in random order

INPUT

```
i 40
i 30
e
m
i 8
d 0 7
d 1 10
e
i 6
i 80
i 24
i 35
e
e
m
e
s
```

OUTPUT

```
30
40
7
6
10
24
24
```

4_extract minimum from empty queue

INPUT

```
e
e
i 30
i 10
i 79
i 5
m
i 2
e
e
s
```

OUTPUT

```
-1
-1
```

```
5
2
5
```

5_Decrease key to an already present value

INPUT

```
i 35
i 5
i 18
d 0 5
m
i 77
e
e
e
e
s
```

OUTPUT

```
5
5
18
35
77
```

6_ All elements have equal priority

INPUT

```
i 2
i 2
i 2
d 0 2
i 2
i 2
d 4 2
m
e
e
m
e
e
e
s
```

OUTPUT

```
2
2
2
2
2
2
2
```

7_Some elements have equal priority

INPUT

```
i 16
i 14
m
i 10
```

```
i 7
e
i 14
i 16
e
e
e
e
e
s
```

OUTPUT

```
14
7
10
14
14
16
16
```

8_Find minimum multiple times

INPUT

```
i 17
i 19
i 7
d 1 15
d 2 14
m
m
m
m
e
s
```

OUTPUT

```
7
7
7
7
7
```

9_Inserted and emptied

INPUT

```
i 17
i 19
i 7
d 1 15
d 2 14
i 20
i 18
i 40
m
e
e
e
e
e
e
```

```
e
i 25
i 19
i 4
e
e
e
s
```

OUTPUT

```
7
7
14
15
18
20
40
-1
4
19
25
```

10_Priority queue with large values

INPUT

```
i 5001
i 2020
i 1505
i 100010
i 999999
i 40005
d 0 1400
m
d 4 888909
e
e
i 70000
e
e
m
e
e
e
e
e
s
```

OUTPUT

```
1400
1400
2020
5001
40005
70000
70000
100010
888909
-1
-1
```
