Assignment 6: Binary heaps

- 1. For the given data as integers perform following operations in min-heap:
 - a. Create heap
 - b. Display heap elements in sequence after k deletions of root element

```
Input: (T, n, k, {x_i})
```

```
2
11
4
12 1 21 2 24 23 15 26 4 33 10
7
2
5 2 8 1 4 6 10

Output:
1 2 15 4 10 23 21 26 12 33 24
12 24 15 26 33 23 21
1 2 6 5 4 8 10
4 5 6 8 10
```

- 2. Implement max-heap and perform insertion and deletion Operations.
 - 1. Insert the element
 - 2. Delete the element
 - 3. Display all elements
 - 4. Quit

Input: (n, x_i)

```
1 30
1 50
1 70
2 10
2 50
1 100
3
```

Output:

inserted inserted inserted 10 not found deleted 100 30 70

3. Implement heapsort.

```
Input: (T, n_i, {x_i})
```

```
2
6
12 11 20 5 16 7
8
15 24 16 22 5 20 40 8
Output:
5 7 11 12 16 20
```

5 8 15 16 20 22 24 40

- 4. Implement binary heap using a binary tree (not arrays). Binary heap operations are:
 - 1. Insert
 - 2. Delete min
 - 3. Check full
 - 4. Check empty
 - 5. Quit

Input: (T, n, k, {x_i})

```
10
1 24
1 6
1 28
1 5
1 63
1 19
1 94
2
2
2
2
2
2
2
4
5
```

Output:

5

```
24
6 24
6 24 28
5 6 24 28
5 6 28 24 63
5 6 19 24 63 28
5 6 19 24 63 28 94
6 24 19 94 63 28
19 24 28 94 63
24 63 28 94
28 63 94
63 94
94
Empty
underflow
true
```

5. For a given array of elements, determine the minimum number of interchanges needed to convert it into a max-heap.

```
Input: (T, n, {x_i})
2
13
89 19 50 17 12 15 2 5 7 11 6 9 100
8
15 24 16 22 5 20 40 8

Output:
3
```

6. Given a sequence find if it constitutes a max-heap.

Input: (T, n_i, {x_i})

```
3
7
10 8 6 4 5 2 1
7
10 5 6 4 8 2 1
7
5 2 8 1 4 6 10
```

Output:

yes no no

7. Write a program to construct priority queue using heap. Print the final contents of the priority queue.

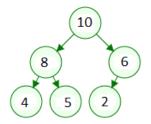
Input: (n, {x_i})

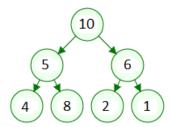
```
8
15 24 32 2 5 28 48 16
Output:
48 16 32 15 5 24 28 2
```

8. Construct a heap structure implemented not with an array but as a normal complete binary tree with with elements given in the following order. (Figure for the first two test cases is given below). Find if it constitutes a max-heap.

Input: (T, n_i, {x_i})







Output:

yes no no