

A group of students wish to make use of stacks so as to sort a given sequence of distinct single digit integers, i.e. $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. They decided to make use of two stacks, say $S1$ and $S2$. At the start, they pushed all the numbers onto $S1$. Then each time they pop the topmost number from $S1$ and push it onto $S2$ using an algorithm.

The algorithm they designed says, push the topmost number of $S1$ onto $S2$ if either $S2$ is empty or the topmost number of $S2$ is less than the topmost number of $S1$. The algorithm works in association with the following constraints:

- Topmost numbers of both the stacks are single digit integers.

Let the topmost number of $S1$ is x and the topmost number of $S2$ is y . If $|x - y| = 1$, then pop x as well as y from both the stacks and push a two digit number onto $S2$. The two digit number is formed using x and y such that the digit at tens place is greater than that present at ones place.

- Topmost number of $S1$ is a single digit and that of $S2$ is a double digit integer.

Let the topmost number of $S1$ is x and the topmost number of $S2$ is $ab = a * 10 + b$.

- If $|a - x| = 1$ or $|b - x| = 1$, then pop x as well as ab from both the stacks and push a two digit number onto $S2$. The two digit number is formed using two out of three digits (a , b , and x) such that the digit at tens place is the greatest and that present at ones place is the smallest among the three digits.
- If $x > a$, then pop x from $S1$ and push it onto $S2$, i.e. push a single digit number x over a double digit number ab .

- Topmost number of $S1$ is a double digit and that of $S2$ is a single digit integer.

Let the topmost number of $S1$ is $ab = a * 10 + b$ and the topmost number of $S2$ is x . If $|a - x| = 1$ or $|b - x| = 1$, then pop x as well as ab from both the stacks and push a two digit number onto $S2$. The two digit number is formed using two out of three digits (a , b , and x) such that the digit at tens place is the greatest and that present at ones place is the smallest among the three digits.

- Topmost numbers of both the stacks are double digit integers.

Let the topmost number of $S1$ is $ab = a * 10 + b$ and the topmost number of $S2$ is $cd = c * 10 + d$. If $|a - d| = 1$ or $|b - c| = 1$, then pop ab as well as cd from both the stacks and push a two digit number onto $S2$. The two digit number is formed using two out of four digits (a , b , c , and d) such that the digit at tens place is the greatest and that present at ones place is the smallest among the four digits.

- If the topmost number of $S2$ is greater than the topmost number of $S1$ and none of the above listed constraints is satisfied then pop and store the topmost number of $S1$ in a temporary space, say $temp$. Pop from $S2$ and push onto

$S1$ till the topmost number of $S2$ is either greater than $temp$ or none of the above listed constraints (between $temp$ and the topmost number of $S2$) is satisfied.

At the end, $S1$ becomes empty and contents of $S2$ will give the sorted sequence in coded form which has to be decoded.

Input:

- Line 1 contains an integer N , the total number of single digit integers.
- Line 2 contains N integers separated by space which are to be sorted.

Output:

- Line 1 is a space separated contents of $S2$.
- Line 2 is the decoded space separated sorted sequence.

Sample Input I:

```
6
4 1 3 6 5 2
```

Sample Output I:

```
64 31
6 5 4 3 2 1
```

EXPLANATION I:

Contents of both the stacks will vary in the following manner:

```
S1: 4 1 3 6 5 2 (top)
S2: EMPTY

S1: 4 1 3 6 5
S2: 2

S1: 4 1 3 6
S2: 2 5

S1: 4 1 3
S2: 2 65

S1: 4 1 65
S2: 2

S1: 4 1 65
S2: 32

S1: 4 1
S2: 32 65

S1: 4 65
S2: 32

S1: 4 65
S2: 31

S1: 4
S2: 31 65

S1: EMPTY
S2: 31 64 (top)

So the output is 64 31. And the sorted sequence will be 6 5 4 3 2 1.
```

Sample Input II:

```
6
5 8 0 3 9 6
```

Sample Output II:

98	65	3	0
9	8	6	5
3	0		

EXPLANATION II:

Contents of both the stacks will vary in the following manner:

S1: 5 8 0 3 9 6 (top)
S2: EMPTY
S1: 5 8 0 3 9
S2: 6
S1: 5 8 0 3
S2: 6 9
S1: 5 8 0 9
S2: 6
S1: 5 8 0 9 6
S2: EMPTY
S1: 5 8 0 9 6
S2: 3
S1: 5 8 0 9
S2: 3 6
S1: 5 8 0
S2: 3 6 9
S1: 5 8 9
S2: 3 6
S1: 5 8 9 6
S2: 3
S1: 5 8 9 6 3
S2: EMPTY
S1: 5 8 9 6 3
S2: 0
S1: 5 8 9 6
S2: 0 3
S1: 5 8 9
S2: 0 3 6
S1: 5 8
S2: 0 3 6 9
S1: 5
S2: 0 3 6 98
S1: 98
S2: 0 3 6
S1: 98
S2: 0 3 65
S1: EMPTY
S2: 0 3 65 98 (top)
So the output is 98 65 3 0. And the sorted sequence will be 9 8 6 5 3 0.