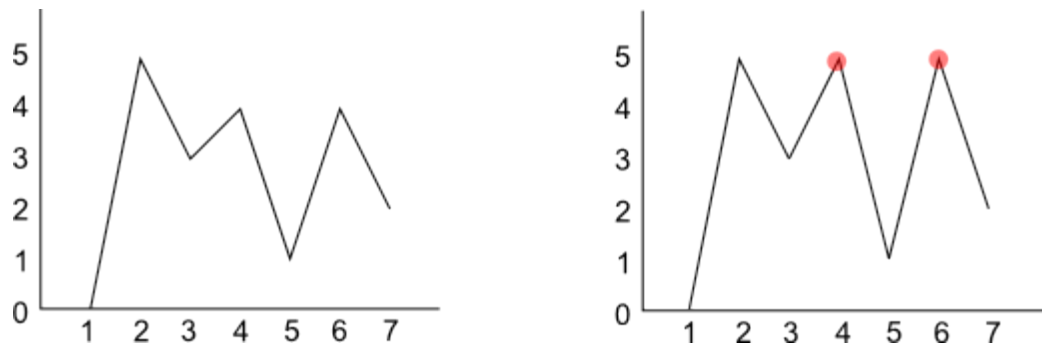
X. Mountain Scenery

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Bolek has found a picture with n mountain peaks painted on it. The n painted peaks are represented by a non-closed polyline, consisting of $2n$ segments. The segments go through $2n + 1$ points with coordinates $(1, y_1)$, $(2, y_2)$, ..., $(2n + 1, y_{2n+1})$, with the i-th segment connecting the point (i, y_i) and the point $(i + 1, y_{i+1})$. For any even i $(2 \le i \le 2n)$ the following condition holds: $y_{i-1} < y_i$ and $y_i > y_{i+1}$.

We shall call a vertex of a polyline with an even x coordinate a *mountain peak*.



The figure to the left shows the initial picture, the figure to the right shows what the picture looks like after Bolek's actions. The affected peaks are marked red, $k = 2$.

Bolek fancied a little mischief. He chose exactly k mountain peaks, rubbed out the segments that went through those peaks and increased each peak's height by one (that is, he increased the y coordinate of the corresponding points). Then he painted the missing segments to get a new picture of mountain peaks. Let us denote the points through which the new polyline passes on Bolek's new picture as $(1, r_1)$, $(2, r_2)$, ..., $(2n + 1, r_{2n+1})$.

Given Bolek's final picture, restore the initial one.

Input

The first line contains two space-separated integers n and k $(1 \le k \le n \le 100)$. The next line contains $2n + 1$ space-separated integers $r_1, r_2, ..., r_{2n+1}$ $(0 \le r_i \le 100)$ — the y coordinates of the polyline vertices on Bolek's picture.

It is guaranteed that we can obtain the given picture after performing the described actions on some picture of mountain peaks.

Output

Print $2n + 1$ integers $y_1, y_2, ..., y_{2n+1}$ — the y coordinates of the vertices of the polyline on the initial picture. If there are multiple answers, output any one of them.

Examples

input Copy

```
3 2
0 5 3 5 1 5 2
```

output Copy

```
0 5 3 4 1 4 2
```

input Copy

```
1 1
0 2 0
```

output Copy

```
0 1 0
```