Farmer John's N $(1 \le N \le 5 \cdot 10^5)$ cows are lined up in a circle. The ith cow has a bucket with integer capacity a_i $(1 \le a_i \le 10^9)$ liters. All buckets are initially full.

Every minute, cow i will pass all the milk in their bucket to cow $i+1$ for $1 \le i < N$, with cow N passing its milk to cow 1. All exchanges happen simultaneously (i.e., if a cow has a full bucket but gives away x liters of milk and also receives x liters, her milk is preserved). If a cow's total milk ever ends up exceeding a_i, then the excess milk will be lost.

After each of $1, 2, \ldots, N$ minutes, how much total milk is left among all cows?

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains N.

The next line contains integers $a_1, a_2, \ldots, a_N$.

OUTPUT FORMAT (print output to the terminal / stdout):

Output N lines, where the i-th line is the total milk left among all cows after i minutes.

SAMPLE INPUT:

```
6
2 2 2 1 2 1
```

SAMPLE OUTPUT:

```
8
7
6
6
6
6
```

Initially, the amount of milk in each bucket is
$[2, 2, 2, 1, 2, 1]$.

- After 1 minute, the amount of milk in each bucket is $[1, 2, 2, 1, 1, 1]$ so the total amount of milk is 8.
- After 2 minutes, the amount of milk in each bucket is $[1, 1, 2, 1, 1, 1]$ so the total amount of milk is 7.
- After 3 minutes, the amount of milk in each bucket is $[1, 1, 1, 1, 1, 1]$ so the total amount of milk is 6.
- After 4 minutes, the amount of milk in each bucket is $[1, 1, 1, 1, 1, 1]$ so the total amount of milk is 6.
- After 5 minutes, the amount of milk in each bucket is $[1, 1, 1, 1, 1, 1]$ so the total amount of milk is 6.
- After 6 minutes, the amount of milk in each bucket is $[1, 1, 1, 1, 1, 1]$ so the total amount of milk is 6.

SAMPLE INPUT:

```
8
3 8 6 4 8 3 8 1
```

SAMPLE OUTPUT:

```
25
20
17
14
12
10
8
8
```

After 1 minute, the amount of milk in each bucket is $[1, 3, 6, 4, 4, 3, 3, 1]$ so the total amount of milk is 25.

SAMPLE INPUT:

```
10
9 9 10 10 6 8 2 1000000000 1000000000 1000000000
```

SAMPLE OUTPUT:

```
2000000053
1000000054
56
49
42
35
28
24
20
20
```

SCORING:

- Inputs 4-5: $N \leq 2000$
- Inputs 6-8: $a_i \leq 2$
- Inputs 9-13: All a_i are generated uniformly at random in the range $[1, 10^9]$.
- Inputs 14-23: No additional constraints.