Problem B
Black Box
Time Limit: 1.0 Seconds

The following **Python**-like pseudo code for function **BlackBox()** takes a list of positive integers and shuffles the integers in the list in a specific way, and returns the result as a list.

Three list methods are used below; For a list `L`, `len(L)` returns the number of items in `L`. `L.append(x)` adds the item `x` to the end of `L`. `L.pop(idx)` removes the item at the specified index `idx` from the list `L` and returns the removed item.

Given a list **Z** of positive integers, write a program to reconstruct a list **I** such that **Z = BlackBox(I)**.

```
function BlackBox( Banana ):
    if len( Banana ) <= 4 :
        exit("Too small Banana")
    Apple  = []                  # [] is an empty list
    Mango  = 0
    Papaya = len( Banana )

    while( Papaya >= 2 ) :
        Kiwi = Banana[ Mango ]
        Apple.append( Kiwi )
        Banana.pop( Mango )
        Papaya = Papaya - 1
        Mango = ( Kiwi + Mango - 1 ) % Papaya
    # end of while

    Apple.append( Banana[ 0 ] )
    Pear    = len( Apple ) - 1
    Orange  = Apple[ Pear ]
    Lime    = Apple[ 0 ]
    Coconut = Orange % Pear
    Melon   = Apple[ Coconut ]
    Apple[ 0 ] = Melon
    Apple[ Coconut ] = Lime

    return ( Apple )
# end of function BlackBox
```

Input

Your program is to read from standard input. The first line contains a positive integer n representing the number of positive integers of a list **`Z`**, where $5 \leq n \leq 200{,}000$. The following n lines contain n positive integers of the list **`Z`** returned from **`BlackBox(I)`**; the i-th line contains the i-th integer of the list **`Z`** between 1 and 100,000, both inclusive.

Output

Your program is to write to standard output. Print n integers of the list **`I`** where **`Z = BlackBox(I)`**, one per line; the i-th line should contain the i-th integer of **`I`**.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
13 113 49 68 91 10 179 2 71 78 45 57 10 88	10 113 179 68 57 45 10 2 88 71 49 78 91

Sample Input 2	Output for the Sample Input 2
9 6 8 7 9 5 1 2 4 3	9 8 7 6 5 1 2 3 4