

Problem H. HackShuffle

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 1024 mebibytes

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

Three specific functions are used below. For a list `L`, the function `len(L)` returns the number of items in `L`. The method `L.append(x)` adds the item `x` to the end of `L`. The method `L.pop(idx)` removes the item at the specified position `idx` from the list `L` (counting from zero) and returns the removed item.

Given a list of positive integers `T`, write a program to reconstruct a list `S` such that `T = HackShuffle(S)`.

```
function HackShuffle( Beta ):
  if len( Beta ) <= 4 :
    exit("Too small Beta")
  Alpha = [] # [] is an empty list
  Gamma = 0
  Delta = len( Beta )

  while( Delta >= 2 ) :
    Omega = Beta[ Gamma ]
    Alpha.append( Omega )
    Beta.pop( Gamma )
    Delta = Delta - 1
    Gamma = ( Omega + Gamma - 1 ) % Delta
  # end of while

  Alpha.append( Beta[ 0 ] )
  Pi = len( Alpha ) - 1
  Omicron = Alpha[ Pi ]
  Lambda = Alpha[ 0 ]
  Rho = Omicron % Pi
  Mu = Alpha[ Rho ]
  Alpha[ 0 ] = Mu
  Alpha[ Rho ] = Lambda

  return ( Alpha )
# end of function HackShuffle
```

Input

The first line contains an integer n ($5 \leq n \leq 200\,000$), the length of list `T`.

The following n lines contain integers T_0, T_1, \dots, T_{n-1} , one per line: the list `T` returned from `HackShuffle(S)` ($1 \leq T_i \leq 100\,000$).

Output

Print n lines containing integers S_0, S_1, \dots, S_{n-1} , one per line, where `S` is a list such that `T = HackShuffle(S)`.

Examples

standard input	standard output
13	10
113	113
49	179
68	68
91	57
10	45
179	10
2	2
71	88
78	71
45	49
57	78
10	91
88	
9	9
6	8
8	7
7	6
9	5
5	1
1	2
2	3
4	4
3	