

# Problem E: Cookie selection 2

## Algorithms for Programming Contests

### Restrictions

Time: 2 seconds

Memory: 512 MB

### Problem description

As chief programmer at a cookie production plant you have many responsibilities, one of them being that the cookies produced and packaged at the plant adhere to the very demanding quality standards of the Nordic Cookie Packaging Consortium (NCPC).

At any given time, your production line is producing new cookies which are stored in a holding area, awaiting packaging. From time to time there are also requests from the packaging unit to send a cookie from the holding area to be packaged. Naturally, you have programmed the system so that there are never any packaging requests sent if the holding area is empty. What complicates the matter is the fact that representatives of the NCPC might make a surprise inspection to check if your cookies are up to standard. Such an inspection consists of the NCPC representatives demanding that the next few cookies sent to the packaging unit instead be handed over to them; if they are convinced that these cookies look (and taste) the same, you pass the inspection, otherwise you fail.

Since you do not have the time to always be on the lookout for cookie craving NCPC inspectors, you need there to be an automatic mechanism that decides which cookie should be sent to packaging. Fortunately, a while ago the production plant invested in a measurement thingamajig capable of measuring cookie diameters with a precision of 1 nanometer (nm). Back then you started to always send out the cookie of median size, hoping it would please them, but unfortunately this often lead to them complaining about the cookies being too small, so now you finally decided to adjust that mechanism. From now on, instead of the median it should always send the *largest* cookie from the holding area that is *matched or exceeded* in size by *at least a third* of the holding area's current cookie population to the packaging unit on request.

## Input

Each line of the input contains either a positive integer  $d$ , indicating that a freshly baked cookie with diameter  $d$  nm has arrived at the holding area, or the symbol '#', indicating a request from the packaging unit to send a cookie for packaging. There are at most 600 000 lines of input, and you may assume that the holding area is empty when the first cookie in the input arrives to the holding area. Furthermore, you have read somewhere that the cookie oven at the plant cannot produce cookies that have a diameter larger than 30 centimeters (or  $3 \cdot 10^8$  nm).

## Output

A sequence of lines, each indicating the diameter in nm of a cookie sent for packaging, in the same order as they are sent.

## Sample input and output

Input	Output
1	5
2	4
3	3
4	6
5	10
6	
#	
#	
#	
#	
10	
#	