Task: ARC
Architects

XVI OI, Stage II, Day two. Source file `arc.*` Available memory: 32 MB. *12.02.2009*

King Byteasar decided to have a new palace erected. Thus he invited the architects into a competition for the best palace design. To encourage them to work quickly, he announced that the projects would be reviewed in the order of submission.

As the king's commission is very prestigious, the architects from all over the world submitted their works to the Royal House. In fact, the number of designs that arrived is so large that Byteasar has no hope to review them all by himself. Therefore he asked his chancellor to do an initial reviewing of the projects according to the following rules:

- the chancellor is to choose only k projects, thereby ultimately rejecting the remaining ones, as Byteasar knows he will only be able to review at most k projects himself.
- the chosen projects will be presented to Byteasar in their submission order, as such was the king's promise;
- the chancellor is to choose **the best** set of k projects, with respect to the following definition.
  We shall sometimes refer to the set of projects as a sequence of projects, as they can be naturally ordered by their submission time. We say that the sequence of projects $(p_1, p_2, \ldots, p_k)$ is **better** then the sequence $(r_1, r_2, \ldots, r_k)$, if for some $l \geqslant 1$ the first $l-1$ projects in both sequences are equally good, and the l-th project of the sequence p is strictly better than the l-th project of the sequence r, i.e. $p_i = r_i$ for $i < l$, and $p_l > r_l$.

The designs are still arriving, and the king has not yet specified any deadline. The chancellor would like to begin his work soon, but he is afraid of Byteasar's legendary anger, should he make a mistake. Hence he asked for your help.

Write a programme that:

- using a supplied library reads the number k and the sequence of numbers denoting the quality of successive designs,
- determines the best (with respect to aforementioned rules) subsequence of k designs,
- returns the subsequence of qualities, again using the same library.

Library Usage

To use the library, one needs to put the following in his/her programme:

- C/C++: `#include ''carclib.h''`
- Pascal: `uses parclib;`
- Java: no additions in the code are required, but to run the solution, one needs the compiled library jarclib (`jarclib.class` file) in the same directory as the programme.

The library supplies three procedures, functions, or static methods:

- `inicjuj` — returns an integer k ($1 \leqslant k \leqslant 1\,000\,000$), denoting the number of projects the resultant sequence should contain. This one should be used exactly once, at the very beginning of the programme's execution.
  - C/C++: `int inicjuj();`
  - Pascal: `function inicjuj(): longint;`
  - Java: `public static int inicjuj();`, which is a static method of the `jarclib` static class.
- `wczytaj` — its i-th call returns an integer p_i ($1 \leqslant p_i \leqslant 1\,000\,000\,000$) denoting the quality of the i-th project (the larger the number, the better the project), or 0, which means there are no more projects. The number of projects is not known in advance, but you may assume that there are at least k and no more than $15\,000\,000$ projects. This function should be called until there are no more projects, and **not once thereafter**.
  - C/C++: `int wczytaj();`
  - Pascal: `function wczytaj(): longint;`
  - Java: `public static int wczytaj();`, which is a static method of the `jarclib` static class.

- `wypisz` — with this procedure/function the qualities of successive projects to be presented to the king are printed out. This one should be used exactly k times; in its i-th call it should be given the quality of i-th project (in the required order). The k-th call of this procedure/function will terminate your programme.
  - C/C++: `void wypisz(int jakoscProjektu);`
  - Pascal: `procedure wypisz(jakoscProjektu: longint);`
  - Java: `public static void wypisz(int jakoscProjektu);`, which is a static method of the `jarclib` static class.

You are not allowed to open any files, nor use standard input and output in your programme. The solution will be compiled together with the library with the following commands:

- C: `gcc -O2 -static carclib.c arc.c -lm`
- C++: `g++ -O2 -static carclib.c arc.cpp -lm`
- Java: `javac arc.java`, and the compiled `jarclib` library file — `jarclib.class` — will be in the same directory.
- Pascal: `ppc386 -O2 -XS -Xt arc.pas`, and the `parclib` library file will be in the same directory.

You will find exemplary libraries and solutions demonstrating their use in the ??? directory. The exemplary library reads a test scenario from the standard input, formatted as follows:

- The first line of the standard input contains one positive integer k.
- Each of the following input lines contains one positive integer; the $(i+1)$-th line contains the number p_i, denoting the quality of the i-th submitted project.
- The last input line contains the number 0, which signifies the end of the projects list.

The exemplary library prints out k lines to the standard output — the qualities of the projects registered by the programme.

Exemplary execution of the programme

C/C++	**Pascal**	**Java**	**Returned values and commentary**
`k = inicjuj();`	`k := inicjuj();`	`k = jarclib.inicjuj();`	From now on $k = 3$.
			Reading the qualities of successive projects.
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	12
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	5
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	8
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	3
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	15
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	8
`wczytaj();`	`wczytaj();`	`jarclib.wczytaj();`	0 — signifies the end of the sequence
			Printing out the solution. It is a sequence of 3 elements:
`wypisz(12);`	`wypisz(12);`	`jarclib.wypisz(12);`	12
`wypisz(15);`	`wypisz(15);`	`jarclib.wypisz(15);`	15
`wypisz(8);`	`wypisz(8);`	`jarclib.wypisz(8);`	8