

блокчето има само всяко k -то дете, и то само по веднъж, т.е. след като си вземе от шоколада, детето излиза от кръга.

Да се дефинира функция `josephus :: [a] -> (Int -> [a])`, която получава като аргумент списък от идентификатори на деца и връща едноместна функция на естествено число k , която връща списък от картончетата (идентификаторите) на децата в последователността, в която те ще излязат от кръга (която е еквивалентна на техния ред на отчупване).

Примери:

```
(josephus [1,2,3,4,5,6,7]) 3 → [3,6,2,7,5,1,4]
```

```
(josephus [1,2,3,4,5,6,7]) (-1) → error "k was not natural"
```

```
(josephus [1,2,3,4,5,6,7,8,9,10]) 1 → [1,2,3,4,5,6,7,8,9,10]
```

```
(josephus [1,2,3,4,5,6,7,8,9,10]) 2 → [2,4,6,8,10,3,7,1,9,5]
```

```
(josephus "fpFMIsu") 4 → "MfsIuFp"
```

Обяснение за пример 1:

```
[1,2,3,4,5,6,7] => начало
```

```
[1,2,4,5,6,7]   => 3 си отчупва и излиза от кръга [3]
```

```
[1,2,4,5,7]     => 6 си отчупва и излиза от кръга [3,6]
```

```
[1,4,5,7]       => 2 си отчупва и излиза от кръга [3,6,2]
```

```
[1,4,5]         => 7 си отчупва и излиза от кръга [3,6,2,7]
```

```
[1,4]           => 5 си отчупва и излиза от кръга [3,6,2,7,5]
```

```
[4]             => 1 си отчупва и излиза от кръга [3,6,2,7,5,1]
```

```
[]             => 4 си отчупва и излиза от кръга [3,6,2,7,5,1,4]
```