

Stos naleśników

Mały Jasio jest bardzo głodny. Na szczęście ma przed sobą stos bardzo gorących naleśników. Chciałby je ułożyć w jakiejś sensownej kolejności, więc zdecydował się na posortowanie ich względem średnic tak, że najmniejszy naleśnik jest na wierzchu stosu, a największy na samym dole stosu. W ten sposób poczeka aż wystygną i będzie mógł od razu zacząć je szybko wcinać. Problem w tym, że nie może dotknąć żadnego naleśnika - inaczej by się poparzył. Ma jednak łypatkę, która umożliwia mu jeden ruch - może ją wsunąć pod naleśnik o numerze x , po czym przetrząść część stosu od wierzchołka do x włącznie, na drugą stronę. Czyli inaczej mówiąc odwrócić kolejność ułożenia naleśników znajdujących się między wierzchołkiem stosu oraz x włącznie. Operację tą **flip(x)** definiujemy podając numer naleśnika x . Naleśniki numerujemy kolejno liczbami od **1** do **n**, gdzie **n** oznacza ilość naleśników, zaczynając od naleśnika na wierzchołku stosu.

Twoim zadaniem jest napisać program, który dla danego ułożenia początkowego stosu naleśników, wypisuje sekwencję argumentów dla których kolejno wywołujemy operację **flip** w celu posortowania stosu według zamiaru Jasia. Jako, że Jasiowi się nie śpieszy, ponieważ trochę czasu minie zanim wszystkie naleśniki wystygną, a możliwości posortowania jest wiele to możesz wypisać dowolną sekwencję ruchów, która zrobi to o co prosi Jaś.

Wejście

Wejście rozpoczyna liczba testów $t \leq 10$. Każdy test składa się z liczby naleśników $n \leq 1000$ po której następuje dwukropek i n liczb naturalnych, oddzielonych spacjami, z przedziału **[1; 500]** oznaczających średnice kolejnych naleśników. Naleśniki podawane są w naturalnej kolejności od pierwszego numeru do n-tego (dla przypomnienia - od wierzchołka do naleśnika na samym dole);

Wyjście

Dla każdego testu należy wypisać sekwencję numerów naleśników dla których kolejno wykonujemy operację **flip** w celu posortowania stosu, zakończoną zerem. Sędzia zaakceptuje rozwiązanie, jeśli po zakończeniu wykonywania tych operacji stos będzie posortowany, tak jak chce tego Jasio.

Przykład

Wejście:

```
1  
5: 5 1 2 3 4
```

Przykładowe wyjście:

```
5 4 0
```

Objaśnienie do przykładu: sekwencja ruchów: flip(5), flip(4) przekształca stos w tej kolejności:
5 1 2 3 4 -> 4 3 2 1 5 -> 1 2 3 4 5.