

Nadmiar trudu

Nadmiar trudu

Za siedmioma bitami, za siedmioma bajtami i jednym terabajtem, w malowniczo położonej krainie, wzdłuż drogi co jeden kilometr rozstawione są w porządku rosnącym punkty widokowe. Chcąc odwiedzić każdy z tych punktów i przebyć jak najkrótszą drogę można rozpocząć od pierwszego, a skończyć na ostatnim. Ale jak zrobić to najgorzej, czyli tak, aby pokonana droga była jak najdłuższa?

Dla zadanej liczby punktów widokowych musisz ustalić jak długa będzie taka droga i zaplanować ją w taki sposób, aby porządek odwiedzanych punktów był leksykograficznie najmniejszy.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita d ($d \leq 100$) - liczba przypadków testowych.

W kolejnych d wierszach dla każdego przypadku testowego podana jest liczba punktów widokowych n ($2 \leq n \leq 10^6$).

Dla danych wejściowych zachodzi warunek: $d \cdot \max(n) \leq 10^6$.

Wyjście

Dla każdego przypadku testowego w pierwszym wierszu należy podać długość najdłuższej drogi, w wierszu drugim leksykograficznie najmniejszą permutację ciągu liczbowego $1, 2, \dots, n$ składającego się na kolejność odwiedzanych punktów widokowych tak, aby pokonana droga spełniała warunek najdłuższej.

Przykład

Wejście

```
1
5
```

Wyjście

```
11
2 4 1 5 3
```

Wyjaśnienie do przykładu

Rozpoczynamy w punkcie widokowym nr 2, dalej odwiedzamy kolejno punkty widokowe: 4, 1, 5 i kończymy w punkcie widokowym nr 3. Tak zaplanowana trasa jest jedną z najdłuższych i wynosi $2 + 3 + 4 + 2 = 11$, a wypisany ciąg jest leksykograficznie najmniejszy.