

Bez błędne sortowanie

XVIII OIJ, zawody III stopnia – dzień drugi
28 kwietnia 2023

Kod zadania: **sor**
Limit czasu: **8 s**
Limit pamięci: **256 MB**



Bajtusia wróciła z wakacji i postanowiła uporządkować kolekcję zdjęć, które zrobiła podczas swojego wyjazdu. W tym celu nadała nowe nazwy plikom ze zdjęciami, tak żeby były to kolejne liczby naturalne. Pierwsze zdjęcie wakacji dostało więc nazwę 1.jpg, kolejne zdjęcie nazwę 2.jpg i tak dalej. Po wykonaniu mozolnej pracy nazwania plików w ten sposób, usunęła niektóre zdjęcia, które wydały jej się mało interesujące.

Niestety, niedługo później odkryła, że jej system operacyjny wyświetla pliki w złej kolejności. W katalogu, w którym pozostawiła zdjęcia 1.jpg, 2.jpg oraz 10.jpg, są one wyświetlane w niewłaściwej kolejności: po nazwie 1.jpg, pojawia się 10.jpg, zaś 2.jpg jest dopiero na końcu. Bajtusię bardzo zrytowała ta sytuacja, ale po chwili zrozumiała, co się stało: system sortuje nazwy plików nie według wartości liczbowych lecz alfabetycznie porównując nazwy znak po znaku aż do napotkania pierwszej różnicy. Wcześniej na liście jest plik z nazwą, która na tej pozycji ma znak o mniejszym kodzie ASCII. Znak kropki (.) ma kod ASCII 46, zaś kolejne cyfry otrzymały kody od 48 (dla zera) do 57 (dla dziewiątki). Porównując więc nazwy 10.jpg oraz 2.jpg, pierwsza różnica występuje już na pierwszej pozycji, a znak 1 ma mniejszy kod ASCII niż znak 2.

Bajtusia nie jest zbyt chętna, żeby ponownie podjąć olbrzymi wysiłek zmiany nazw swoich plików. Wpadła na prostsze rozwiązanie: usunie pewną niewielką liczbę zdjęć tak, aby kolejność pozostawionych była prawidłowa. Pomóż jej ustalić ile najwięcej zdjęć może pozostawić, żeby ich kolejność według systemu operacyjnego była zgodna z uporządkowaniem według rosnącej kolejności numerów.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N ($1 \leq N \leq 200\,000$) określająca liczbę zdjęć, które Bajtusia zdecydowała się pozostawić po wcześniejszym odrzuceniu zdjęć nieciekawych. W drugim (ostatnim) wierszu wyjścia znajduje się ciąg N parami różnych liczb naturalnych A_i ($1 \leq A_i \leq 10^{18}$), pooddzielanych pojedynczymi odstępami. Są to numery zdjęć Bajtosi, które pozostawiła Bajtusia podane w dowolnej kolejności. Zdjęcia mają więc nazwy $A_1.jpg, A_2.jpg, \dots, A_N.jpg$.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna nieujemna liczba całkowita R – maksymalna liczba zdjęć, którą Bajtusia może pozostawić w katalogu, aby kolejność zdjęć była zgodna z jej oczekiwaniami.

W drugim (ostatnim) wierszu wyjścia powinien się znaleźć ciąg R parami różnych liczb naturalnych (w dowolnej kolejności), określający numery zdjęć, które Bajtusia powinna pozostawić.

Jeżeli istnieje wiele możliwych rozwiązań, Twój program może wypisać dowolne z nich.

Ocenianie

Jeżeli tylko pierwszy wiersz wyjścia jest prawidłowy, Twój program otrzyma 50% punktów za test, pod warunkiem, że zakończy się bez błędu wykonania i nie przekroczy dostępnych limitów czasu i pamięci. Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
$N = 2$	10
$N \leq 20$	30
pozostawione zdjęcia Bajtosi mają numery 1, 2, 3, ..., N	40
$N \leq 1\,000$	58
wszystkie $A_i \leq 500\,000$	72



Przykłady

Wejście dla testu sor0a:

```
3
2 10 1
```

Wyjście dla testu sor0a:

```
2
10 1
```

Wyjaśnienie do przykładu: Innymi poprawnymi odpowiedziami będą również 1 10 (bo kolejność wypisanych numerów nie ma znaczenia) oraz 1 2 (bo możemy odrzucić zdjęcie 10 zamiast 2), a także 2 1.

Wejście dla testu sor0b:

```
12
1 2 3 4 5 6 7 8 9 10 11 12
```

Wyjście dla testu sor0b:

```
9
1 2 3 4 5 6 7 8 9
```

Wejście dla testu sor0c:

```
9
1 10 100 2 20 200 3 30 300
```

Wyjście dla testu sor0c:

```
5
300 30 3 2 1
```

Pozostałe testy przykładowe

- test sor0d: $N = 90$, test zawiera wszystkie liczby dwucyfrowe w przypadkowej kolejności,
- test sor0e: $N = 200\,000$, test zawiera kolejne liczby naturalne począwszy od 10^{17} .