

Ivica je nedavno naučio efikasan način pretrage brojeva. Naime, za svaki poredani niz brojeva potrebno je puno manji broj koraka za pronalazak bilo kojeg broja nego što je to potrebno kada je niz nasumično poredan. Metoda koju Ivica proučava zasniva se na smanjivanju niza za polovicu u svakom koraku pretrage. Tako primjerice ako su cijeli brojevi u intervalu od 1 do 100 uključivo, prvo će zadani broj tražiti na sredini intervala, ako ga tamo ne pronađe, interval će podijeliti na dva dijela te provjeriti da li se traženi broj nalazi u intervalu [1,49] ili [51,100] te ovisno o tome gdje se nalazi – odbaciti će onaj interval u kojem se sigurno ne nalazi i nastaviti s pretragom u preostalom intervalu.

Ivica je primjetio da su neki brojevi skloniji tome da budu brzo pronađeni, dok su drugi na samim krajevima po pitanju potrebnog broja koraka za pronalazak. Ivicu zanima koji brojevi će se sigurno moći pronaći u točno K koraka pretrage – za svaki broj $K < P$, gdje P predstavlja najveći broj mogućih uzastopnih prepolavljanja radnog intervala.

Ulazni podaci:

- Cijeli broj N: $1 \leq N \leq 50000$;
- N cijelih brojeva X_i : $-100\ 000 \leq X_i \leq 100\ 000$

Izlazni podaci:

- P redaka izlaza, gdje P označava najveći broj koraka za **opisani način pretrage**:
 - o Cijeli broj K: broj koraka pretrage
 - o Znak dvotočja s po jednim razmakom prije i poslije
 - o Uzlazno poredani niz brojeva T_i : svi brojevi do kojih se dolazi u točno K koraka pretrage; brojeve razdvojiti točno jednim znakom razmaka

Sustavna ograničenja:

- 1 sekunda za izvršavanje po test primjeru
- 32 MB HEAP memorija, 32 MB STACK memorija

Test primjeri:

U L A Z	5 1 2 3 4 5	6 10 20 30 40 50 60	12 9 3 4 10 7 5 8 6 11 12 2 1
	1 : 3 2 : 1 5 3 : 2 4	1 : 40 2 : 20 60 3 : 10 30 50	1 : 7 2 : 3 10 3 : 1 5 8 12 4 : 2 4 6 9 11

Naziv datoteke: **distribucija.c** ili **distribucija.cpp**

Također su dozvoljene i ekstenzije za BASIC (.bas) i Pascal (.pas) ukoliko radite u jednom od tih jezika.