

Twin Cookies

| Zadatak | Twin Cookies |
|------------------------|------------------|
| Ulazni podaci | standardni ulaz |
| Izlazni podaci | standardni izlaz |
| Vremensko ograničenje | 1 sekunda |
| Memorijsko ograničenje | 256 MiB |

Ovo je interaktivni zadatak. Vaš program treba komunicirati s programom izrađenim od strane povjerenstva pišući poruke na standardni izlaz i čitajući poruke sa standardnog ulaza.

Sofija priprema rođendansku proslavu za svoje blizance. Blizanci obožavaju kekse. Za rođendan žele ostvariti svoju najveću želju, isprobati kekse iz posebne selekcije Tvornice Keksa Jedinstvenog Okusa (TKJO).

Okus svakog keksa proizvedenog u TKJO tvornici označen je prirodnim brojem između 1 i 10^{16} (uključivo). Budući da su blizanci vrlo ljubomorni, svaki od njih mora dobiti skup kekasa takav da je suma vrijednosti okusa kekasa u oba skupa jednaka.

TKJO prima samo narudžbe od **točno** n kekasa. U svakoj narudžbi kupac specificira vrijednost okusa svakog od n kekasa koje želi naručiti.

Kao što i samo ime tvornice kaže, TKJO neće poslati dva kekasa s istom vrijednošću okusa istom kupcu. Odnosno, Sofija mora osigurati da ne naruči dva kekasa s istom vrijednosti okusa (niti u jednoj, niti u dvije različite narudžbe). Budući da Sofija prije nije naručivala iz TKJO, može naručiti najviše jedan keks svih ponuđenih vrijednosti okusa.

Postoji više prepreka na Sofijinom putu. Naime, opće je poznato da TKJO ima užasnu kurirsku službu. Kada neki klijent naruči n kekasa, samo jedan od tih n kekasa zapravo stigne do klijenta. Preostale kekse pojedju zaposlenici tvrtke prije no što stignu do klijenta. Klijent također ne može utjecati na to koji će n kekasa zapravo stići do njega.

Budući da se rođendan blizanaca opasno približava, Sofija ima vremena napraviti najviše 101 narudžbu. Pomozite joj.

Preciznije, od vas tražimo da:

1. Najprije, naručite kekse. Možete napraviti najviše 101 narudžbu, svaka se sastoji od točno n unaprijed određenih vrijednosti okusa. Možete napraviti najviše jednu narudžbu istovremeno. **Odmah nakon što napravite narudžbu bit će vam poznata vrijednost okusa keksa kojeg ćete dobiti.**

Zapamtite da ne možete iskoristiti vrijednost okusa keksa u više od jedne narudžbe. (Primjerice, iako ste naručili keks s vrijednosti okusa t , a taj keks nije dostavljen, ubuduće **ne možete** naručiti keks s vrijednošću okusa t).

2. Potom, podijelite kekse. Nakon što imate dovoljno keksa, potrebno je **neke** od njih dodijeliti blizancima. Oba blizanca trebaju dobiti barem jedan keks, a suma vrijednosti okusa koju je dobio svaki blizanac mora biti jednaka. **Ne morate podijeliti sve kekse koje ste dobili.**

Izlazni podaci

Svaki puta kada vaš program ispiše jednu ili više linija na standardni izlaz, potrebno je napraviti **flush** izlaznog toka podataka. Ovo je potrebno kako biste osigurali da vaš izlaz dođe do programa kojeg je napisalo povjerenstvo.

Ispravni primjeri **flusha** izlaza u podržanim programskim jezicima:

- U jeziku C++ imate više opcija.
 - `fflush(stdout);`
 - `std::cout << std::flush;`
 - `std::cout << std::endl;` (primijetite da ovo ispisuje i znak za prazan redak)
 - čitanje pomoću `std::cin` također flusha izlazni tok podataka.
- U jeziku Java, možete koristiti `System.out.flush()`
- U jeziku Python, možete koristiti `sys.stdout.flush()`

Interakcijski protokol

Vaš program treba odraditi sljedeći slijed akcija:

1. Učitati vrijednost n sa standardnog ulaza.
2. Najviše 101 puta:
 1. Najprije, ispisati u jedan redak narudžbu od n keksa na standardni izlaz.
 2. Zatim, učitati vrijednost okusa keksa kojeg vam je dostavila kurirska služba. Garantirano je da se ta vrijednost nalazi među n vrijednosti koje ste naveli u vašoj narudžbi.
3. Ispisati tri retka koja označavaju jedan valjan način podjele nekih dostavljenih keksi blizancima.

Grader će ispisati svaki cijeli broj u zaseban redak.

Da biste naručili kekse, ispišite jedan redak koji počinje sa znakom ? nakon kojeg slijedi n cijelih brojeva: vrijednosti okusa keksa koje želite naručiti. Prije svakog od n brojeva ispišite jedan znak razmaka.

Zapamtite da možete napraviti najviše 101 narudžbu i da ne možete istu oznaku okusa iskoristiti u više od jedne narudžbe.

Nakon što ste naručili dovoljno keksa, ispišite konačna tri retka koja predstavljaju valjanu podjelu keksa blizancima.

Prvi od ta tri retka treba biti oblika " $m k$ " sa $m, k > 0$: broj keksa koje će dobiti prvi i drugi blizanac.

U drugom se retku treba nalaziti m brojeva odvojenih razmakom: vrijednosti okusa keksa koje će dobiti prvi blizanac.

Analogno, u trećem se retku treba nalaziti k brojeva odvojenih razmakom: vrijednosti okusa keksa koje će dobiti drugi blizanac.

Sljedeća svojsvat trebaju biti zadovoljena:

1. Svaki blizanac treba dobiti barem jedan keks
2. Svaki blizanac treba dobiti kekse jednake sume vrijednosti okusa.
3. Samo kekci koji su uspješno dostavljeni smiju biti iskorišteni.
4. Svaki od dostavljenih keksa smije biti dodijeljen najviše jednom blizancu.

Svaki izlaz koji zadovoljava ova ograničenja bit će prihvaćen. Također, kekci mogu biti ispisani u proizvoljnom poretku.

Nakon što ispišete posljednja tri retka, napravite flush izlaza i **regularno završite izvođenje svog programa.**

Bodovanje

Podzadatak 1 (8 bodova): $n = 1$

Podzadatak 2 (9 bodova): $1 \leq n \leq 2$

Podzadatak 3 (18 bodova): $1 \leq n \leq 25$

Podzadatak 4 (16 bodova): $1 \leq n \leq 200$

Podzadatak 5 (13 bodova): $1 \leq n \leq 1000$

Podzadatak 6 (36 bodova): $1 \leq n \leq 5000$

Primjeri

| Ulaz | Izlaz |
|------|--------|
| 1 | ? 13 |
| 13 | ? 7 |
| 7 | ? 31 |
| 31 | ? 12 |
| 12 | ? 5 |
| 5 | ? 3 |
| 3 | ! 2 3 |
| | 7 13 |
| | 12 5 3 |
| 2 | ? 3 7 |
| 7 | ? 2 8 |
| 2 | ? 1 5 |
| 5 | ! 2 1 |
| | 2 5 |
| | 7 |

Pojašnjenje primjera

Primjeri ulaza i izlaza trebaju se čitati redak po redak. Vaš program mora alternirati između čitanja jedne vrijednosti sa standardnog ulaza i pisanja jednog (ili tri na kraju) retka na standardni izlaz.

Program izrađen od strane organizatora (grader) proizvoljno odabire koji će keks biti dostavljen. Odnosno, grader u nekim testovima može biti adaptivan, a ponekad može vratiti uniformno nasumičan keks (strategija gradera vam nije poznata). Primjerice, grader vam neće nužno vratiti iste kekse kao što je prikazano u drugom probnom primjeru.