

## Piškotki za dvojčka

Ime naloge	Piškotki za dvojčka
Vhod	standardni vhod
Izhod	standardni izhod
Časovna omejitev	1 sekunda
Omejitev spomina	256 MB

*To je interaktivna naloga. Tvoj program bo komuniciral z ocenjevalnim sistemom z izmeničnim pisanjem sporočil na standardni izhod in branjem sporočil iz standardnega vhoda.*

Sophie pripravlja rojstnodnevno zabavo za svoja dvojčka. Dvojčka obožujeta piškote. Za rojstni dan bi rada poskusila nekaj novega: piškote, ki jih pečejo pri Unique Cookie Tastiness Company (UCTC).

Vsak piškot, spečen pri UCTC ima celoštevilsko okusnost med vključno 1 in  $10^{16}$ . Dvojčka sta ljubosumna drug na drugega, zato mora vsak od njiju prejeti piškote z enako skupno vsoto okusnosti.

UCTC sprejema samo naročila **natančno**  $n$  piškotov. V vsakem naročilu stranka pove okusnost za vsakega od  $n$  piškotov, ki jih želijo.

Zaradi zvestobe svojemu nazivu Unique Cookie Tastiness Company zavrača peko dveh piškotov z enako okusnostjo za isto stranko. Sophie mora paziti, da nikoli ne naroči iste okusnosti dvakrat -- ne v istem naročilu, niti v dveh različnih naročilih ne. Sophie še nikoli ni kupovala pri UCTC, zato lahko naroči vsako okusnost enkrat.

Še ena ovira je pred Sophie: Dostavni sistem UCTC je grozen. Kadar stranka naroči  $n$  piškotov, samo eden od teh  $n$  pride do stranke. Ostale zaposleni v dostavni službi pojejo po poti. Stranka nima vpliva na to katerega izmed  $n$  naročenih piškotov bo na koncu tudi dobila.

Rojstni dan se hitro približuje in Sophie ima čas za največ 101 naročil. Tvoja naloga je, da ji pomagaš.

Natančneje, narediti moraš:

1. Najprej, naroči piškote. Opraviš lahko največ 101 naročil, vsako naj ima natančno

$n$  željenih okusnosti. Naenkrat lahko opraviš eno naročilo. **Takoj po vsakem naročilu, izveš okusnost piškota, ki ga dejansko prejmeš.**

Upoštevaj, da iste okusnosti ne moreš uporabiti večkrat, niti v različnih naročilih ne. (Če naročiš piškot z okusnostjo  $t$ , a ni dostavljen, **ne smeš** še enkrat naročiti piškota z okusnostjo  $t$ .)

2. Potem razdeli piškote. Ko imaš dovolj piškotov, jih **nekaj** razdeli med dvojčka. Vsak od njiju naj prejme vsaj en piškot, skupna vsota okusnosti piškotov pa naj bo enaka za oba dvojčka. **Ni ti potrebno porabiti vseh piškotov, ki jih prejmeš!**

## Izhod

Vsakič, ko tvoj program izpiše eno ali več vrstic na standardni izhod, moraš **zaključiti izpis (flushing the output stream)**. To je potrebno zato, da podatki, ki jih izpišeš, takoj pridejo do ocenjevalnega sistema.

Primeri:

- V C++, imaš več možnosti:
  - `fflush(stdout);`
  - `std::cout << std::flush;`
  - `std::cout << std::endl;` (opomba: to izpiše tudi znak za novo vrstico)
  - branje iz `std::cin` tudi zaključi izpis.
- v Javi, lahko uporabiš `System.out.flush()`
- v Pythonu, lahko uporabiš `sys.stdout.flush()`

## Protokol za interakcijo

Tvoj program naj izvede naslednje zaporedje operacij:

1. Preberi število  $n$  iz standardnega vhoda.
2. Največ 101-krat:
  1. Najprej, izpiši eno vrstico, ki vsebuje naročilo  $n$  piškotov, na standardni izhod.
  2. Potem, preberi okusnost piškota, ki ga prejmeš, iz standardnega vhoda. Zagotovo bo to število iz nabora  $n$  vrednosti, ki so bile v trenutnem naročilu.
3. Izpiši tri vrstice, ki predstavljajo en veljaven način delitve nekaj piškotov med dvojčka.

Ocenjevalni sistem bo vsako število izpisal v svojo vrstico.

Za naročilo piškotov izpiši eno samo vrstico z  $?$ , ki mu sledi  $n$  celih števil: okusnosti piškotov, ki jih želiš naročiti. Pred vsakim izmed  $n$  celih števil izpiši en presledek.

Zapomni si, da lahko opraviš največ 101 naročil in da nobene okusnosti ne smeš uporabiti večkrat.

Ko naročiš dovolj piškotov, izpiši končne tri vrstice, ki opisujejo katere piškote naj Sophie podari dvojčkoma.

Prva od teh vrstic naj ima obliko " $m k$ ", kjer sta  $m, k > 0$  števili piškotov, ki naj jih prejmeta prvi in drugi dvojček.

V drugi vrstici naj bo  $m$  celih števil ločenih s presledki: okusnost piškotov, ki jih prejme prvi dvojček.

Podobno naj bo v tretji vrstici  $k$  celih števil ločenih s presledki: okusnost piškotov, ki jih prejme drugi dvojček.

Izhod mora zadostiti naslednjim pogojem:

1. Vsak dvojček mora prejeti vsaj en piškot.
2. Vsak dvojček mora prejeti piškote z enako skupno vsoto okusnosti.
3. Uporabiš lahko samo piškote, ki jih dejansko prejmeš.
4. Vsak piškot lahko podariš največ enemu od dvojčkov.

Vsak izhod, ki zadosti tem pogojem, bo sprejet kot pravilen. Izbrane piškote lahko izpišeš v katerem koli vrstnem redu.

Ko izpišeš zadnje tri vrstice, še zadnjič zaključi izpis (flush the output stream) in potem **normalno zaključi svoj program**.

## Ocenjevanje

Podnaloga 1 (8 točk):  $n = 1$

Podnaloga 2 (9 točk):  $1 \leq n \leq 2$

Podnaloga 3 (18 točk):  $1 \leq n \leq 25$

Podnaloga 4 (16 točk):  $1 \leq n \leq 200$

Podnaloga 5 (13 točk):  $1 \leq n \leq 1000$

Podnaloga 6 (36 točk):  $1 \leq n \leq 5000$

## Primeri

vhod	izhod
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

## Opomba

Primeri vhoda in izhoda se berejo vrstica za vrstico. Tvoj program izmenjaje bere eno vrednost iz standardnega vhoda in izpiše eno vrstico (oziroma tri na koncu) na standardni izhod.

Ocenjevalni sistem poljubno izbere, kateri piškot prejmeš. To pomeni, da se ocenjevalni sistem lahko prilagodi tvojim naročilom pri določenih testih, lahko pa povsem naključno izbere piškot pri drugih testih. Na primer, za  $n = 2$ , če narediš enako zaporedje naročil kot v drugem primeru, lahko prejmeš drugačen nabor piškotov.