

Verkefni: ZER

Zeroes and ones



BOI 2025, Dagur 0. Tiltækt minni: 256 MB.

2025.04.25

Það er tvíundaröð a_0, \dots, a_{n-1} . Þú veist ekki stökin, en þú getur beðið um summu af hvaða tveim aðskildum stökum. Verkefnið þitt er að finna röðina án þess að nota of mikið af spurningum.

Interaction

Þetta er gagnvirkir verkefni. Þú skalt skrifa forrit sem finnur réttu lausnina við verkefninu og talar við gagnvirkan dómara. Dómarinn les úttakið frá forritinu þínu og skrifar í inntakið á forritinu þínu.

Dómarinn skrifar fyrst út eina línu með heiltölunni n ($3 \leq n \leq 200\,000$), sem er lengdin á röðinni.

Síðan skal forritið þitt spyrja spurninga á eftirfarandi máta:

- Forritið skal skrifa út eina línu á forminu

$i \ j$

þar sem i og j eru mismunandi heiltölur ($0 \leq i \neq j \leq n-1$). Þessi lína táknar eina spurningu varðandi stökin a_i and a_j .

- Svarið verður gefið sem:

s

þar sem s er heiltala ($0 \leq s \leq 2$) jöfn $a_i + a_j$.

Þegar forritið þitt hefur ákveðið röðina, þá skal það skrifa tvær línur á forminu

n

$a_0 \ a_1 \ \dots \ a_{n-1}$

þar sem n er lengdin af röðinni og a_0, a_1, \dots, a_{n-1} eru stökin í henni. **Síðan skal forritið þitt hætta keyrslu. Ef spurðar eru fleiri spurningar getur útkoman verið Wrong answer.**

Hafa skal í huga að eftir hverja spurningu eða lokasvar þá þarf að flush-a úttaksbiðminninu með því að nota `cout.flush()` (eða `fflush(stdout)` ef notað er `printf`) í C++ eða bæta við `flush = True` frumbreytunni við `print` í Python. Annars gæti forritið þitt fengið `Time Limit Exceeded`.

Forritið þitt má ekki opna skrár, eða nota önnur úrræði. Þú mátt nota villuúttakið til að kemba forritið þitt, en hafðu í huga að skrifa í villuúttakið getur tekið tíma.

C++, inntak/úttak með straum

Þú skalt passa að hafa réttan haus (`#include <iostream>`). Þú skalt enda hverja línu í úttakinu með `std::endl`. Sjáið eftirfarandi dæmi fyrir upphafssamskipti.

```
std::cin >> n;
std::cout << i << ' ' << j << std::endl;
std::cin >> sum;
```

C++, inntak/úttak með því að nota stdio

Þú skalt passa að hafa réttan haus (`#include <stdio.h>`). Eftir að hafa skrifað út hverja línu skal kalla á `fflush(stdout)`. Sjáið eftirfarandi dæmi fyrir upphafssamskipti.

```
scanf("%d", &n);
printf("%d %d\n", i, j);
fflush(stdout);
scanf("%d", &sum);
```

Python

Hvert kall í `print` fallið skal hafa `flush = True` sem frumbreytu. Sjáið eftirfarandi dæmi fyrir upphafssamskipti.

```
n = int(input())
print(f"{i} {j}", flush = True)
sum = int(input())
```

Gagnvirkir sýnidæmi

Aðgerð	Gildi	Úttak	Lýsing
lesa n	–	5	$n = 5$
summa	0 1	1	$a_0 + a_1 = 1$
summa	1 2	1	$a_1 + a_2 = 1$
summa	3 4	2	$a_3 + a_4 = 2$, svo $a_3 = a_4 = 1$
summa	0 3	2	$a_0 + a_3 = 2$, svo $a_0 = 1$, og frekar $a_1 = 0$ og $a_2 = 1$
svar	5 1 0 1 1 1		Rétt svar, $m = 4 \leq n = 5$ spurningar, 100% stig.

Stigagjöf

Undirflokkur	Takmarkanir	Stig
1	$3 \leq n \leq 1000$	50
2	$3 \leq n \leq 200\,000$	50

Dómarinn þarf ekki endilega að festa alla inntaksröðina a_0, \dots, a_{n-1} í byrjun keyrslu. Í staðinn getur hann breytt gildunum í röðinni, svo lengi sem það er í gildi við allar fyrri spurningar, m.ö.o. þá er dómarinn aðlögunarhæfur.

Skilgreinum m sem hámarksfjölda af spurningum sem forritið þitt má spyrja. Sjá má að neðan hvaða prósentuhlutfall lausnin þín fær miðað við fjölda spurninga fyrir hvert prufutilvik:

Queries	Percentage of points
$m \leq n$	100% stig fyrir prufutilvikið
$m = n + 1$	80% stig fyrir prufutilvikið
$n + 1 < m \leq n^2 - n$	50% stig fyrir prufutilvikið
$m > n^2 - n$	0% stig fyrir prufutilvikið (Wrong answer)