

Bankomat albo filantrop (bankomat)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

To zadanie jest interaktywne. Po wypisaniu każdego wiersza należy opróżnić bufor.

W C++ możesz użyć `cout << flush`, w Java – `System.out.flush()`, w Python – `sys.stdout.flush()`.

Należy ściśle trzymać się protokołu opisanego w rozdziale *Protokół interakcji* – niewczytanie wiadomości wysłanej przez interaktor skutkuje werdyktem *wrong answer*.

Wielu uczestników wrocławskich obozów ma zapewne poczucie, że można się na nich wiele nauczyć. Obóz, na którym powstało to zadanie jest dofinansowany z wiadomo czyich podatków (chodzi oczywiście o wszystkich polskich podatników). Niektórzy uczestnicy mogą więc mieć w sobie naturalny odruch, żeby społeczeństwu, które finansuje im lepszą przyszłość się zrewanżować i dać tym podatnikom coś od siebie. Gdyby tak nie marnować pieniędzy podatnika na uczniów, którzy w trakcie konkursu grają w gry? Albo chociaż gdyby najlepsi, którym już udało się spieniężyć swoje ponadprzeciętne umiejętności zdobyte podczas nieprzespanych nocy na obozie, wspólnie postanowili oddać choćby promil swojego majątku na dalszy rozwój talentów...

Tak właśnie pomyślał Karol, który dzięki obozom i startom w Olimpiadzie Informatycznej stał się niewyobrażalnie bogaty. Z dobroci serca postanowił on przekazać swoją fortunę na cele charytatywne (m.in. na wypłaty dla kadry na przyszłych obozach informatycznych). Karol jest jednak wielkim miłośnikiem gotówki i nie da przecież bankom zarabiać na kosztach przelewu (przy dużych kwotach jest to niepomijalna sprawa). Postanowił więc wypłacić swoją fortunę z bankomatu.

W bankomacie znajduje się nieznana Karolowi kwota M bajtalarów. Chciałby wypłacić znaczną część tej kwoty (dokładniej, co najmniej 99% tego co tam jest). Czas Karola jest bardzo cenny, więc musi dokonać mniej niż 70 transakcji, a będzie nawet bardziej zadowolony, jeżeli uda się wypłacić te dużo pieniędzy wykonując co najwyżej 50 transakcji (dostaniesz wtedy więcej punktów).

W każdej transakcji Karol wpisuje na klawiaturze bankomatu kwotę (nieujemną liczbę całkowitą), którą chce wypłacić. Bankomat wypłaca tę kwotę pod warunkiem, że znajduje się w nim co najmniej tyle pieniędzy (wtedy gotówki w kieszeni Karola przybywa, zaś gotówki w bankomacie ubywa). W przeciwnym przypadku transakcja zwraca błąd: bankomat nie wypłaca nic, wymagając ponownego włożenia karty, wprowadzenia PIN-u i oczekiwania na rezultat kolejnej transakcji, co okropnie irytuje Karola. Jeżeli taki błąd nastąpi podczas całej interakcji cztery razy, to irytacja Karola osiągnie taki poziom, że porzuci swój filantropijny pomysł i za pieniądze w kieszeni kupi sobie samochód sportowy (albo lody albo nic, zależnie od tego ile będzie tej gotówki miał). W każdym razie, w takiej sytuacji Ty uzyskasz werdykt *wrong answer* i 0 punktów za test.

Możesz założyć, że Karol ma na koncie znacznie więcej pieniędzy niż jakkolwiek bankomat na świecie (a w szczególności ten, z którego właśnie Karol wypłaca pieniądze) ma dostępnej gotówki.

Twoim zadaniem jest napisać program, który będzie proponował Karolowi kwoty, które powinien wpisywać do kolejnych prób transakcji, tak żeby na cele charytatywne przeznaczył znaczny procent kwoty znajdującej się w bankomacie – patrz sekcja *Punktacja*.

Protokół interakcji

Interakcja składa się z rund zapytanie – odpowiedź interaktora, które kończą się informacją od Twojego programu, który w pewnym momencie proponuje Karolowi, żeby zakończył pracę z bankomatem i przekazał zebraną dotychczas gotówkę na cele charytatywne.

W każdym zapytaniu Twój program powinien wypisać kwotę $K \geq 0$, którą Karol ma wprowadzić na klawiaturze bankomatu.

Wprowadzenie $K = 0$ oznacza, że program sugeruje Karolowi zakończyć sesję przy bankomacie. Należy wtedy zakończyć program (w przeciwnym razie program może otrzymać dowolny werdykt, na przykład inny niż *accepted*).

Wysłanie wartości $K > 0$ oznacza natomiast propozycję, aby Karol poprosił bankomat o wypłatę kwoty K bajtalarów. Należy wtedy wczytać odpowiedź TAK lub NIE, która określa, czy ta transakcja doszła do skutku (jeżeli odpowiedź to TAK, to transakcja doszła do skutku, a więc Karol ma o K bajtalarów więcej

w ręku/kieszeni, zaś dostępna w bankomacie kwota zmniejsza się o K , zaś jeżeli odpowiedź to NIE, to w bankomacie przed transakcją znajdowało się mniej niż K bajtalarów, a więc transakcja nie doszła do skutku). Uzyskanie czwartej odpowiedzi NIE oznacza werdykt `wrong answer`: Twój program powinien się zakończyć bez dalszego wypisywania (w przeciwnym razie może otrzymać dowolny werdykt, oczywiście inny niż `accepted`).

Sprawdzaczka **NIE** jest adaptacyjna. Dla każdego testu jest uczciwie ustalona kwota M (przed rozpoczęciem interakcji).

Punktacja

Niech L oznacza kwotę, którą Karol przekazał na cele charytatywne, zaś T oznacza liczbę udanych transakcji przy bankomacie. Dla przypomnienia: M oznacza kwotę, która początkowo znajdowała się w bankomacie (przed wykonaniem transakcji Karola).

Jeżeli $L < M \cdot 99\%$ lub $T \geq 70$, Twój program 0% punktów za test i werdykt `wrong answer`.

W przeciwnym przypadku liczba punktów zależy od T . Jeżeli $T \leq 50$, otrzymasz 100% punktów za test. Jeżeli zaś $50 < T < 70$, otrzymasz $100 - 5(T - 50)$ procent punktów za test.

Innymi słowy: Twoim celem jest napisać program, który wypłaci z bankomatu co najmniej 99% znajdujące się tam kwoty wykonując co najwyżej 50 udanych transakcji i co najwyżej trzy nieudane. Częściowe punkty przyznawane są rozwiązaniom, które wykonują więcej niż 50 udanych transakcji, ale mniej niż 70 oraz spełniają wszystkie pozostałe warunki (każda udana transakcja ponad limit pięćdziesięciu zabiera pięć punktów procentowych punktacji za test).

Ograniczenia

$$1 \leq M \leq 10^{18}.$$

Grupa	Warunki	Punkty
1	$M \leq 50$	10
2	$M \leq 500$	15
3	$M \leq 5\,000$	20
4	$M \leq 100\,000$	15
5	$M \leq 10^9$	20
6	$M \leq 10^{18}$	20

Przykładowa interakcja

W teście przykładowym $M = 1\,000$.

Wejście	Wyjście
	1100
NIE	
	900
TAK	
	50
TAK	
	100
NIE	
	45
TAK	
	0

W tej interakcji Karolowi udało się dokonać $T = 3$ transakcji i wypłacił z bankomatu kwotę $900 + 50 + 45 = 995$ bajtalarów (czyli ponad 99% kwoty znajdujące się na początku w bankomacie). Za tę interakcję należy się więc pełna punktacja.