

Baloni

WWI 2024 – grupa 3
Dzień Próbny – 14 sierpnia 2024

Kod zadania: **bal**
Limit pamięci: **1 GiB**



Maurycy, jak każda małpka, chciałby tylko jeść banany i uprawiać małpie figle. Niestety, tak sielankowe życie nie było mu dane. Urodził się w krainie *Baloni*, w której wszystkie małpki muszą ciągle walczyć o przetrwanie – w krainie *Baloni* małpki toczą bezustanny bój ze złowieszczymi balonami.

Maurycy nie jest jednak zwykłą małpką – chciałby zbudować lepszy świat dla siebie i swoich rodaków. Bój z balonami jest nieunikniony, jednak da się usprawnić go na tyle, by czas pomiędzy bitwami małpki mogły poświęcić na rozwój swojej cywilizacji.

Bitwa z balonami przebiega następująco: przed każdą małpką znajduje się rząd balonów. Każdy z balonów znajduje się na pewnej wysokości. Małpka może rzucić rzutką na wybraną przez siebie wysokość – taka rzutka leci ciągle na tej samej wysokości, mijając kolejne balony, póki nie natrafi na żaden na wysokości swojego lotu – przy spotkaniu rzutki z balonem, balon natychmiast pęka, a eksplozja związana z jego pęknięciem zrzuca rzutkę o jeden poziom niżej. Rzutka dalej kontynuuje trajektorię na swojej nowej wysokości, aż nie napotka kolejnego balona.

Pomóż Maurycemu zoptymalizować metody rzutów małpek podając mu minimalną sumaryczną liczbę rzutów, które mogą wykonać małpki, aby zbić wszystkie balony.

Biblioteka

Jest to zadanie interaktywne, to znaczy Twój program będzie porozumiewał się z biblioteką. Aby użyć biblioteki, należy załączyć nagłówek `#include "ballib.h"`. Twój program nie powinien czytać ze standardowego wejścia, ani wypisywać na standardowe wyjście.

Biblioteka udostępnia następujące procedury:

- `int liczba_malpek()` – podaje liczbę małpek n , które strzelają do balonów. Możesz założyć, że $1 \leq n \leq 10^5$, a małpki są ponumerowane od 1 do n . Tej funkcji należy użyć dokładnie raz, jako pierwsze wywołanie funkcji z tej biblioteki.
- `int kolejny_balon(int i)` – podaje wysokość h ($1 \leq h \leq 10^9$) kolejnego balonu w rzędzie balonów przed małpką o numerze i . Jeżeli przed tą małpką nie znajduje się więcej balonów lub podana liczba nie znajduje się w przedziale od 1 do n , funkcja zwróci -1 . Tej funkcji możesz używać dowolną liczbę razy, w dowolnym miejscu programu, choć dla każdej wartości i od momentu gdy funkcja po raz pierwszy zwróci -1 będzie zwracać wyłącznie -1 .

Twoim zadaniem jest implementacja procedury:

- `int minimalna_liczba_rzutow()` – zwracającej minimalną sumaryczną liczbę rzutów, które mogą wykonać małpki, aby zbić wszystkie balony.

Plik z rozwiązaniem może zawierać funkcje pomocnicze, lecz nie może zawierać funkcji `int main()`.

Możesz przyjąć, że rozwiązanie będzie kompilowane wraz z biblioteką przy użyciu następujących poleceń:

- **C:** `gcc -O3 -static ballib.cc bal.c -lm`
- **C++:** `g++ -O3 -static -std=c++20 ballib.cc bal.cpp -lm`

Eksperymenty

Na *SIO2*, w dziale *Pliki* znajduje się archiwum o nazwie *bal_dla_zaw.zip* zawierające przykładowe pliki biblioteki i przykładowe, niepoprawne rozwiązanie ilustrujące sposób użycia biblioteki.

Przykładowy przebieg działania funkcji `minimalna_liczba_rzutow()`

Wywołanie	Efekt	Opis
<code>liczba_malpek();</code>	$n := 3$	Są 3 małpki.
<code>kolejny_balon(1)</code>	$h := 8$	Pierwszy balon przy pierwszej małpce znajduje się na wysokości 8.
<code>kolejny_balon(1)</code>	$h := 8$	Drugi balon przy pierwszej małpce znajduje się na wysokości 8.
<code>kolejny_balon(1)</code>	$h := 7$	Trzeci balon przy pierwszej małpce znajduje się na wysokości 7.
<code>kolejny_balon(1)</code>	$h := -1$	Przy pierwszej małpce znajdują się tylko 3 balony.
<code>kolejny_balon(2)</code>	$h := 1$	Pierwszy balon przy drugiej małpce znajduje się na wysokości 1.
<code>kolejny_balon(2)</code>	$h := -1$	Przy pierwszej drugiej znajduje się tylko 1 balon.
<code>kolejny_balon(3)</code>	$h := 6$	Pierwszy balon przy trzeciej małpce znajduje się na wysokości 6.
<code>kolejny_balon(3)</code>	$h := 9$	Drugi balon przy trzeciej małpce znajduje się na wysokości 9.
<code>kolejny_balon(3)</code>	$h := -1$	Przy pierwszej trzeciej znajdują się tylko 2 balony.
<code>return 5;</code>	—	Wszystkie balony da się przebić za pomocą dokładnie 5 strzałów.

Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	Sumaryczna liczba balonów nie przekroczy 20.	4 s	7
2	Sumaryczna liczba balonów nie przekroczy 10^3 .	2 s	17
3	Sumaryczna liczba balonów nie przekroczy 10^6 .	2 s	55
4	Sumaryczna liczba balonów nie przekroczy 10^7 .	12 s	21