Zadanie: KRA
Krążki 2

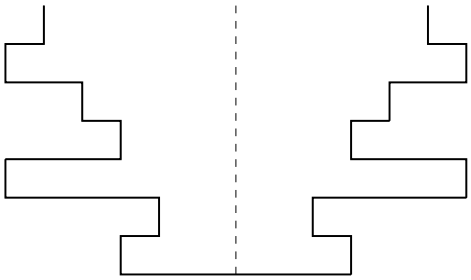
POTYCZKI ALGORYTMICZNE

Potyczki Algorytmiczne 2017, runda próbna. Dostępna pamięć: 256 MB. *16.11.2017*

Uwaga! To jest zadanie *rozproszone*. Zanim zaczniesz je rozwiązywać, zapoznaj się z informacjami dotyczącymi tego typu zadań dostępnymi w serwisie internetowym zawodów.

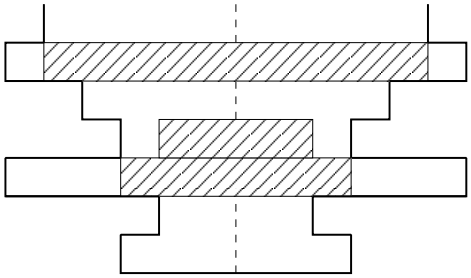
Młody Janek obchodził niedawno swoje 19 urodziny. Z tej okazji rodzice postanowili dać mu prezent nawiązujący do tego sprzed wielu lat*. Tegoroczny prezent to ponownie zabawka składająca się z rurki i krążków, tylko dużo większa.

Janek nie mógł ukryć wzruszenia, przyglądając się znajomym kształtom rurki. Składa się ona z pewnej liczby połączonych walców (o takiej samej grubości) z wyciętymi w środku (współosiowo) okrągłymi otworami różnej średnicy. Rurka jest zamknięta od dołu, a otwarta od góry. Na poniższym rysunku przedstawiono przykładową taką rurkę, złożoną z walców, w których wycięto otwory o średnicach kolejno: 5 dm, 6 dm, 4 dm, 3 dm, 6 dm, 2 dm i 3 dm.



Krążki w zabawce Janka są walcami o różnych średnicach i takiej samej grubości jak walce tworzące rurkę.

Janek przypomniał sobie również swoją ulubioną zabawę z dzieciństwa, na którą poświęcił tyle zimowych wieczorów. Mając do dyspozycji pewien zestaw krążków, zastanawiał się zawsze, na jakiej głębokości zatrzymałby się ostatni z nich, gdyby wrzucał je kolejno do rurki centralnie (czyli dokładnie w jej środek). Jako głębokość zatrzymania krążka rozumiemy tu numer (od góry) walcowego otworu, w którym ten krążek się zatrzyma. Dla przykładu, gdyby wrzucić do powyższej rurki krążki o średnicach kolejno 3 dm, 2 dm i 5 dm, to otrzymalibyśmy następującą sytuację, w której ostatni krążek zatrzymał się na głębokości 2.



Jak widać, każdy kolejny krążek po wrzuceniu spada, dopóki się nie zaklinuje (czyli nie oprze się o walec, w którym wycięty jest otwór o mniejszej średnicy niż średnica krążka) albo nie natrafi na przeszkodę w postaci innego krążka lub dna rurki.

Janek przypomniał sobie również, jak duże trudności sprawiała mu ta zabawa, gdy był mały. Na szczęście pomógł mu wtedy pewien wybitny programista – jego program zawsze szybko wyznaczał głębokość, na której zatrzymałby się ostatni krążek. Obecnie Janek jest już dużo mądrzejszy, ale nadal trochę niecierpliwy. Znalazł stary program, ale okazało się, że dla nowej zabawki działa on niedopuszczalnie długo. Dlatego wynajął klaster obliczeniowy i zwrócił się do Ciebie z prośbą o napisanie nowego programu.

Dane wejściowe

Twój program nie może używać standardowego wejścia. Zamiast tego dane testowe zostaną mu udostępnione za pośrednictwem dostarczonej biblioteki interaktywnej. Aby jej użyć, wpisz w swoim programie w języku C++:

*O tamtych pamiętnych urodzinach możesz przeczytać pod adresem `http://main.edu.pl/pl/archive/oi/13/kra`

```
#include "krazki.h"
```

Biblioteka udostępnia cztery funkcje:

- `PipeHeight()` – zwraca n ($1 \leq n \leq 2 \cdot 10^8$) – wysokość rurki (liczbę walców wchodzących w jej skład).
- `NumberOfDiscs()` – zwraca m ($1 \leq m \leq 2 \cdot 10^8$) – liczbę krążków, które Janek zamierza wrzucić do rurki.
- `HoleDiameter(i)` – zwraca r_i ($1 \leq r_i \leq 10^{18}$) – średnicę otworu wyciętego w i-tym ($1 \leq i \leq n$) od góry walcu tworzącym rurkę.
- `DiscDiameter(j)` – zwraca k_j ($1 \leq k_j \leq 10^{18}$) – średnicę krążka, który zostanie wrzucony do rurki jako j-ty ($1 \leq j \leq m$).

Udostępnionym funkcjom odpowiadają następujące deklaracje w języku C++:

```
int PipeHeight();
int NumberOfDiscs();
long long HoleDiameter(long long);
long long DiscDiameter(long long);
```

W dziale *Pliki* w systemie SIO2 znajduje się archiwum zawierające przykładowe pliki biblioteki oraz (niepoprawne) rozwiązania ilustrujące sposób jej użycia.

Wyjście

Jedyny wiersz wyjścia powinien zawierać jedną liczbę całkowitą, oznaczającą głębokość zatrzymania się ostatniego krążka. Jeżeli krążek ten w ogóle nie wpadnie do rurki, to odpowiedzią powinna być liczba 0.

Komunikacja

Podczas oceny Twojego programu system sprawdzający uruchomi jednocześnie wiele jego instancji, każdą na osobnym komputerze. Instancje powinny komunikować się za pomocą biblioteki `message`. W tym celu w programie w języku C++ należy umieścić wiersz:

```
#include "message.h"
```

Instrukcja dotycząca używania tej biblioteki jest dostępna w serwisie internetowym zawodów w zakładce *Zadania rozproszone*.

Ograniczenia

- Liczba dostępnych węzłów: 100.
- Limit czasu na jeden test: 20 sekund.
- Liczba wiadomości wysłanych przez pojedynczą instancję nie może przekroczyć 1000.
- Sumaryczny rozmiar wiadomości wysłanych przez jedną instancję nie może przekroczyć 96KB.
- Wywołanie dowolnej z funkcji bibliotecznych trwa średnio nie dłużej niż 0.6 mikrosekundy.
- Podany limit pamięci obowiązuje dla pojedynczego węzła.

Przykładowy przebieg programu

Dla przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
`PipeHeight();`	3
`NumberOfDiscs();`	2
`HoleDiameter(1);`	4
`HoleDiameter(2);`	3
`HoleDiameter(3);`	6
`DiscDiameter(1);`	3
`DiscDiameter(2);`	2

poprawnym wynikiem jest:

```
2
```

Testy przykładowe

Po wysłaniu rozwiązania od razu poznasz wyniki Twojego programu na poniższych testach, uruchamianych na 80 komputerach. Limit czasu na jeden test wynosi 4 sekundy, a pozostałe limity są takie same, jak w przypadku właściwych testów.

- `0`: test przykładowy.
- `0a`: test odpowiadający rysunkowi z treści zadania. Odpowiedzią jest 2.
- `0b`:
  - $n = 2 \cdot 10^8$, $m = 2 \cdot 10^8 - 1$,
  - $r_i = 2 \cdot 10^8 + 1 - i$,
  - $k_j = \max(1, j - 1)$.

  Odpowiedzią jest 2.