# Zadanie: SPR
# Sprawdzian

Olimpiada Informatyczna

**ONTAK 2021, dzień czwarty. Dostępna pamięć: 512 MB. Limit czasu: 25 s.** *01.07.2021*

Bajtosz jest nauczycielem matematyki w szkole podstawowej. Należy on do grona pedagogów, którzy w trakcie semestru pozwalają sobie na długie wykłady oraz rozległe dygresje w trakcie odpowiedzi na pytania uczniów. Jego podopieczni bardzo lubią taki tryb prowadzenia zajęć, ale w takim systemie ich wiedza nie jest regularnie sprawdzana (przez brak kartkówek, prac domowych etc.). Pod koniec roku konieczne jest jednak wystawienie ocen za całokształt pracy, co wprowadziło Bajtosza w niemałe zakłopotanie. Wiadomo, że poziom wiedzy każdego z $N$ uczniów można określić przez liczbę całkowitą z przedziału od 0 do $N$ włącznie.

Bajtosz podszedł do sprawy bardzo profesjonalnie i postanowił dokładnie określić poziom wiedzy każdego z uczniów, tak by móc na tej podstawie wystawić ocenę końcową. W tym celu zorganizuje sprawdzian! Niestety ze względów bezpieczeństwa może się on odbyć tylko w formie zdalnej i to za pomocą platformy Beams, która niestety czasami zawodzi swoją funkcjonalnością. Co prawda można organizować sprawdziany na Beams'ach, ale muszą mieć one pewną ściśle zdefiniowaną formę. Bajtosz będzie odkrywał uczniom treści pytań w turach. W danej turze wysyłamy każdemu uczniowi z osobna zadanie o wybranym dla tego ucznia poziomie. Poziom trudności każdego zadania jest liczbą całkowitą z przedziału domkniętego $[0, 2N]$. Jeżeli poziom trudności tego zadania jest mniejszy lub równy od poziomu wiedzy ucznia, to uczeń ten udzieli poprawnej odpowiedzi. W przeciwnym wypadku nie będzie on w stanie rozwiązać tego zadania.

Niestety w implementacji platformy Beams jest pewien uciążliwy błąd. Po każdej turze Bajtosz dostanie informację tylko o tym ilu sumarycznie uczniów udzieliło poprawnej odpowiedzi (zamiast odpowiedzi dla każdego z uczniów z osobna). Ta wada systemu może bardzo wydłużyć sprawdzian... Na domiar złego zgodnie z kodeksem samorządu szkolnego sprawdzian nie może trwać więcej niż $4N$ tur.

Napisz program, który będzie w kolejnych turach wyznaczał poziomy trudności pytań, a na końcu wypisze poziom wiedzy każdego z uczniów.

## Komunikacja

**To zadanie jest interaktywne**. Należy napisać program, który będzie ustawiał ciągi trudności pytań dla uczniów w poszczególnych turach komunikując się w tym celu z dostarczoną biblioteką, a potem zakomunikuje jaki jest poziom wiedzy każdego z uczniów. **W jednym wywołaniu jest** 10 **sprawdzianów**, czyli program musi powtórzyć procedurę dokładnie 10 razy. Liczba uczniów $N$ jest taka sama dla każdego z tych 10 sprawdzianów.

Do programu napisanego w języku $C++$ należy dołączyć nagłówek `sprlib.hpp` za pomocą instrukcji
`#include "sprlib.hpp"`

Biblioteka udostępnia następujące funkcje:

- `int dajN()` – zwraca liczbę uczniów $N$;
- `int tura(vector<int> vec)` – jedna tura sprawdzianu: zadaje uczniom zapytania o trudnościach określonych przez ciąg `vec` (uczeń $i$-ty dostaje pytanie o trudności `vec[i]`), a zwraca liczbę osób, które udzieliły poprawnej odpowiedzi; funkcja ta nie powinna być wywoływana więcej niż $4N$ razy dla jednego sprawdzianu;
- `void koniec(vector<int> vec)` – tę funkcję należy wywołać raz dla każdego sprawdzianu w teście, zwracając ciąg ocen kolejnych uczniów w parametrze `vec`; spowoduje ona zainicjalizowanie nowego sprawdzianu i odświeżenie limitu wywołań $4N$ funkcji `tura`.

Wywołanie funkcji `tura` więcej niż $4N$ razy, lub zwrócenie niepoprawnego ciągu poziomów wiedzy uczniów za pomocą funkcji `koniec` zostanie potraktowane jako błędna odpowiedź. Możesz założyć, że poziom wiedzy uczniów nie zmienia się w trakcie trwania sprawdzianu.

## Ocenianie

W każdym teście do obsłużenia jest dokładnie 10 sprawdzianów. To znaczy, że Twój program powinien dokładnie 10 razy wywołać funkcję `koniec`. Wywołanie jakiejkolwiek funkcji po dziesiątym wywołaniu funkcji `koniec` zostanie potraktowane jako błędna odpowiedź.



*Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".*

| Podzadanie | Ograniczenia | Punkty |
|---|---|---|
| 1 | $10 \leq N \leq 20$ | 40 |
| 2 | $21 \leq N \leq 28$ | 30 |
| 3 | $29 \leq N \leq 34$ | 30 |

# Przykładowy przebieg programu

**Uwaga:** W podanym poniżej przykładzie znajduje się test z parametrem $N < 10$ (mniejszym od parametrów, które mogą występować w grupach testowych).

| Wywołana funkcja | Zwrócony wynik | Opis |
|---|---|---|
| `dajN()` | 2 | Liczba uczniów w klasie dla pierwszego scenariusza sprawdzianu |
| `tura({2, 1})` | 2 | Zarówno pierwszy jak i drugi uczeń odpowiedzieli poprawnie |
| `tura({2, 2})` | 1 | Tylko jeden z uczniów znał odpowiedź na pytanie. Musiał być to pierwszy z nich (poziom jego wiedzy jest równy 2). Drugi uczeń musi mieć zatem wiedzę na poziomie 1. |
| `koniec({2, 1})` | – | Zwracamy wynik. |
| `tura({2, 1})` | 1 | Początek zapytań dla drugiego sprawdzianu. Tym razem nie musimy już wywoływać funkcji `dajN()`. |
| ... | ... | ... |
| `koniec({0, 2})` | – | Wywołanie funkcji `koniec` dla dziesiątego sprawdzianu. Program powinien teraz zakończyć działanie. |

# Eksperymenty

Przykładowe **błędne** rozwiązania wraz z przykładowymi bibliotekami znajdują się na SIO w zakładce *Pliki*. Biblioteki mogą różnić się zachowaniem od tych na sprawdzaczkach i nie spełniać założeń zadania. Mają one jedynie pokazywać sposób interakcji z programem.

W tym zadaniu polecenie kompilacji jest standardowe. Trzeba tylko zadbać o to, by plik `sprlib.hpp` znajdował się w tym samym folderze co rozwiązanie. Dostarczony jest też przykładowy plik `makefile` generujący plik wykonywalny `spr.e` z pliku `spr.cpp` za pomocą komendy:

```
make
```

Można też alternatywnie użyć komendy:

```
g++ -O3 -static -std=c++17 spr.cpp -o spr.e
```



*Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.*