

# Zadanie: FUT

## Futbol [A]



POTYCZKI ALGORYTMICZNE

Potyczki Algoritmiczne 2018, runda czwarta. Limity: 256 MB, 3 s.

13.12.2018 - 14.12.2018

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

Po porażce w ostatnich mistrzostwach świata, nowy trener futbolowej reprezentacji Bajtocji zapowiedział eksperymenty ze składem drużyny. Na każdy z kolejnych, rozgrywanych co tydzień meczów będzie powoływany inny skład, rekrutujący się spośród  $N$  zarejestrowanych w Bajtocji zawodowych futbolistów. Trwało to będzie tak długo, aż sprawdzone zostaną wszystkie możliwe zestawienia drużyny.

Bajtocki futbol różni się trochę od piłki nożnej, czy futbolu amerykańskiego: na boisko może wybiec dowolna drużyna składająca się z co najwyżej  $K$  zawodników. Można wystawić mniej niż  $K$  graczy, reguły dopuszczają również nie wystawienie nikogo – historia zna nawet mecz, w którym spotkały się dwie drużyny bez żadnego zawodnika\*.

Jako wierny (choć niezbyt cierpliwy) kibic chcesz przeczekać czas eksperymentów. Oblicz, ile meczów (a więc ile tygodni) będą one trwały. Ponieważ może to być duża liczba, wystarczy poznać resztę z dzielenia wyniku przez  $P$  – będziesz po prostu włączał telewizor co  $P$  tygodni i sprawdzał, czy trener ogłasza właśnie koniec rotacji składu i jest szansa na wygranie jakiegoś meczu.

Zwróć uwagę na nietypowy limit na liczbę wysłanych wiadomości oraz na to, że w niektórych testach funkcja `GetN()` działa poprawnie tylko dla komputera 0.

## Komunikacja z biblioteką

Aby móc rozwiązywać zadanie, należy na początku programu dopisać dyrektywę:

```
#include "futbol.h"
```

Dodana biblioteka udostępnia następujące funkcje:

- `int GetN()`; – zwraca liczbę  $N$  futbolistów w Bajtocji (ale patrz: sekcja Podzadania).
- `int GetK()`; – zwraca maksymalną liczbę  $K$  graczy mogących zostać powołanych na jeden mecz.
- `int GetP()`; – zwraca liczbę pierwszą  $P$ .

## Wyjście

Dokładnie jedna instancja powinna wypisać jedną liczbę całkowitą – liczbę możliwych składów (czyli tygodni trwania eksperymentów) modulo  $P$ .

## Ograniczenia

- $0 \leq K \leq N < P \leq 10^9$
- $1 \leq N$
- $P$  jest liczbą pierwszą.
- Każdy test będzie uruchomiony na 100 instancjach komputerów.
- Podany limit pamięci dotyczy pojedynczej instancji.
- **Limit liczby wiadomości wychodzących z pojedynczej instancji: 1.**
- **Limit łącznego rozmiaru wiadomości wychodzących z pojedynczej instancji: 64 KB.**

## Podzadania

W niektórych grupach testów funkcja `GetN()` zwraca  $N$  tylko dla komputera 0, natomiast dla innych komputerów zwrócone jest 0. W pozostałych grupach testów funkcja `GetN()` działa poprawnie dla wszystkich komputerów.

---

\*Mecz skończył się remisem.

## Testy przykładowe

Po wysłaniu rozwiązania od razu poznasz wyniki sprawdzania na poniższych testach przykładowych, uruchomionych na 100 instancjach.

Dla poniższego przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
GetN()	4
GetK()	2
GetP()	999999937

poprawnym wynikiem jest:

11

Natomiast dla poniższego przebiegu programu:

Wywołanie funkcji	Zwrócona wartość
GetN()	10 lub 0
GetK()	3
GetP()	13

poprawnym wynikiem jest:

7

**Wyjaśnienie do drugiego przykładu:** Liczba możliwych składów wynosi 176, ale wynik wypisujemy modulo 13. Zauważ, że w tym przykładzie jedynie instancja o numerze 0 może sprawdzić prawdziwą wartość  $N$ . Wszystkie pozostałe instancje po wywołaniu `GetN()` dostaną wartość 0.

## Eksperymenty

W dziale *Pliki* możesz znaleźć przykładową bibliotekę testującą. Składa się ona z plików `futbol.h` oraz `futbol.cpp`. Znajduje się tam też przykładowy (błędny) program `fut_wrong.cpp` korzystający z interfejsu udostępnianego przez bibliotekę.

Aby skompilować Twój program `fut.cpp` z biblioteką, należy wydać następującą komendę (zakładamy, że `rpa` jest zdefiniowane jak w poradniku do zadań rozproszonych):

```
rpa build --source fut.cpp --library futbol.cpp
```

Tak skompilowany program wczytuje test z przykładowego wejścia. Na wejściu powinny znaleźć się trzy liczby  $N$ ,  $K$  oraz  $P$  oddzielone spacjami. Dla ułatwienia, oba testy przykładowe zostały przetłumaczone do powyższego formatu (pliki `fut0a.in` oraz `fut0b.in` w załączonym archiwum).