Zadanie «Harmonogram» (har)

W pewnym zakładzie produkcyjnym linia produkcyjna jest podzielona na odcinki. Każdy odcinek realizuje jedno zadanie. Wynik pracy jednego odcinka jest konieczny do pracy następnych odcinków. Przykładowo: zanim zostanie zbudowane nadwozie samochodu, trzeba polakierować karoserię i trzeba przywieźć z huty szyby. Zanim zostanie polakierowana karoseria musi być zabezpieczona galwanicznie. Zanim karoseria zostanie zabezpieczona galwanicznie musi być wytłoczona z blachy w tłoczni. Inżynier produkcji musi przygotować harmonogram prac. Technolodzy dostarczyli mu pary zadań: zadanie wcześniejsze i zadanie późniejsze. Nie oznacza to, że zadanie późniejsze musi być wykonane bezpośrednio po zadaniu wcześniejszym – mogą między nimi być jeszcze inne zadania. Inżynier produkcji ma za zadanie uszeregować zadania w ciąg (czyli stworzyć harmonogram), taki że dla każdej pary zadań, zadanie wcześniejsze jest wykonane wcześniej niż zadanie późniejsze. Można to zrobić na wiele sposobów.

Proszę napisać program, który będzie znajdował liczbę wszystkich poprawnych harmonogramów zadań modulo 999999937.

Specyfikacja wejścia

Dane wejściowe zawarte są w pliku tekstowym. Zadania są ponumerowane. W pliku wejściowym najpierw podana jest liczba par zadań (jedna liczba w linii). Następnie podane są pary zadań. Para zadań (wcześniejsze i późniejsze) zapisana jest w jednej linii pliku wejściowego.

Specyfikacja wyjścia

W pliku wynikowym należy podać wyłącznie jedną liczbę: liczbę możliwych poprawnych harmonogramów zadań modulo 999999937.

Przykładowe wejście

```
9

1 3
2 4
2 5
6 2
5 3
8 7
```

```
4 3
7 1
8 6
```

Przykładowe wyjście

Możliwe są cztery harmonogramy: `87162543`, `87162453`, `86254713`, `86245713`. Plik wynikowy będzie zawierał tylko jedną liczbę:

```
4
```

Ustalenia techniczne

1. Rozwiązaniem zadania są:

   - program konsolowy napisany w języku C/C++ – Kod źródłowy programu powinien być zawarty wyłącznie w jednym pliku o nazwie `har.c` (dla języka C) lub `har.cpp` (dla języka C++). W pierwszej linii pliku źródłowego należy umieścić w komentarzu *indywidualny kod uczestnika* (IKU). Nie jest dopuszczalne umieszczanie w kodzie jakichkolwiek innych danych umożliwiających zidentyfikowanie uczestnika (także we właściwościach pliku).
   - pliki wyjściowe wypracowane przez program dla danych testowych dostarczonych wraz z treścią zadania – Pliki muszą być nazwane zgodnie z niżej umieszczonym nazewnictwem. Pliki muszą być zgodne ze specyfikacją wyjścia.

   Wszystkie powyższe pliki należy spakować do pliku `IKU-har.zip`, gdzie `IKU` jest *indywidualnym kodem uczestnika*.

2. Program powinien odczytywać dane wejściowe z pliku o nazwie podanej w treści zadania, a wynik należy zapisać też do pliku, którego nazwa jest podana w treści zadania.

3. Należy przyjąć, że dane wejściowe mają poprawny format (opisany w treści zadania). Plik wyjściowy powinien mieć format opisany w treści zadania.

4. W programach można korzystać wyłącznie ze standardowych bibliotek języka C/C++.

5. W programach nie można korzystać z rozwiązań i mechanizmów nieprzenośnych (np. zależnych od systemu operacyjnego).

6. Programy nie mogą:

- tworzyć nowych procesów lub wątków,
- uruchamiać innych programów,
- używać funkcji sieciowych (np. `socket`, `send` itp.),
- oczekiwać na interakcję użytkownika.

7. Zadanie należy przesłać przez stronę konkursu «Złoty Indeks» Platformy Zdalnej Edukacji `https://platforma.polsl.pl/rd/course/view.php?id=7` korzystając z łącza do przesyłania rozwiązań zadania «har».

8. Zadanie jest oceniane w skali 0-15 punktów.

Nazewnictwo plików

Należy ściśle przestrzegać nazewnictwa plików wynikowych. Każdemu plikowi wejściowemu odpowiada odpowiada jeden plik wyjściowy:

plik wejściowy	plik wyjściowy
`har-01.in`	`har-01.out`
`har-02.in`	`har-02.out`
`har-03.in`	`har-03.out`
`har-04.in`	`har-04.out`
`har-05.in`	`har-05.out`
`har-06.in`	`har-06.out`
`har-07.in`	`har-07.out`
`har-08.in`	`har-08.out`
`har-09.in`	`har-09.out`
`har-10.in`	`har-10.out`
`har-11.in`	`har-11.out`
`har-12.in`	`har-12.out`
`har-13.in`	`har-13.out`
`har-14.in`	`har-14.out`
`har-15.in`	`har-15.out`