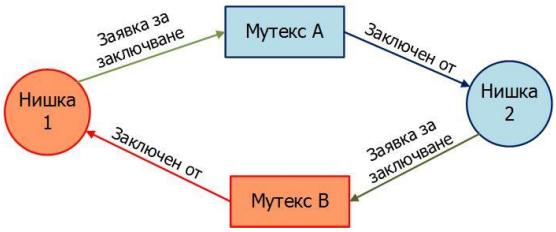
НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА
Русе, 7-9 юни 2019 г.
Група D, 6 клас

Задача D1. МУТЕКСИ



Съвременните езици за програмиране позволяват създаване на програми, които се състоят от няколко нишки на изпълнение. Създава се впечатление, че няколко програми работят паралелно в едно и също адресно пространство, като имат достъп до едни и същи променливи. Ако "многонишковата програма" се изпълнява на компютър с едноядрен процесор и всяка команда се изпълнява за едно и също време, процесорът изпълнява последователно по една команда от всяка нишка, като спазва реда в който нишките са били въведени. Понякога се налага нишките да бъдат синхронизирани помежду си. Например, може да се наложи една нишка да изчака друга, която да приключи някакви изчисления и да съхрани резултата в някаква променлива. Най-простият инструмент за синхронизиране на нишки е мутекс (mutex).

Мутексът е специален обект, който може да е в заключено или отключено състояние. Когато е заключен, мутексът винаги е собственост на точно една нишка (която го е заключила). Има две операции, които една нишка може да приложи към мутекс: **`HOLD`** и **`RELEASE`**. Те се означават съответно със съкращенията HLD и RLS.

Когато дадена нишка приложи **`HOLD`** към мутекс, който в момента е отключен, мутексът се заключва и нишката придобива собственост върху него. Ако една нишка се опита да приложи **`HOLD`** към мутекс, който вече е заключен от друга нишка, нишката остава блокирана докато мутексът бъде отключен.

Когато дадена нишка приложи **`RELEASE`** към мутекс, притежаван от нея, мутексът се отключва. Ако има повече нишки, чакащи да заключат мутекса, една от тях успява да го заключи и получава собственост върху него. Ако има няколко нишки, чакащи да заключат даден мутекс, който е заключен, отключва го тази, която първа получава управлението.

Един от най-често срещаните проблеми в многонишковите програми са „мъртвите хватки" (deadlocks). „Мъртва хватка" възниква когато две или повече нишки се изчакват една друга да освободят мутекс и никоя от тях не може да продължи. Проблемът възниква и когато нишката чака за заключване на мутекс, който е бил заключен от друга нишка, която е приключила без да освободи мутекса.

Ако са дадени описанията на няколко нишки, напишете програма **`mutexes`**, която да изпълнява нишките и да решава дали могат да възникнат блокировки („мъртви хватки") в процеса на тяхното изпълнение. Всяка от нишките представлява последователност от команди във формат:

HLD <име на мутекс>

RLS <име на мутекс>

Командите отговарят на следните изисквания:

1. Имената на мутексите са еднобуквени − главните букви A...Z;
2. Нишките не се опитват да заключат мутекс, който вече притежават;
3. Нишките не се опитват да отключват мутекс, който не притежават;
4. Командите във всяка нишка са номерирани с поредни номера, започвайки от 1.

Вход. Първият ред на стандартния вход съдържа естественото число M, равно на броя на нишките. Следват M блока, описващи последователно всяка една от нишките. Първият ред на всеки един блок, описващ нишката i, съдържа броя на командите в тази нишка N_i и е следван от N_i реда, всеки от които съдържа по една команда. Командите не съдържат допълнителни интервали.

Изход. Изпълнението на командите, включени в нишките, продължава до момента, в който нито една от нишките не може да продължи да се изпълнява. Това се случва ако нишките са приключили нормално или са блокирани, за да изчакат отключването на заключен мутекс. След настъпването на този момент, на стандартния изход се извеждат *M* реда. Всеки от тях описва последователно състоянието на всяка една от нишките, както следва:

− Ако нишка `i` е приключила нормално се извежда текстът:

`"Thread <i> completes successfully."`, където <i> се замества със съответния номер на нишката.

− Ако нишка `i` изпълнява команда номер `j`, като изчаква освобождаването на мутекс *S*, се извежда текстът: **`"Thread <i> waits on command <j> for mutex <S>."`**, където **`<i>`** и **`<j>`** се заместват със съответните им стойности, а **`<S>`** се заменя с името на мутекса.

Описанията на състоянието на нишките са подредени във възходящ ред на техните номера.

Следват още толкова реда, колкото са мутексите, останали заключени след приключване на изпълнението: Например, ако мутекс Y е останал заключен от нишка 2, се отпечатва следния текст:

`"Mutex Y is occupied by thread 2."`

Заключените мутекси се изброяват в азбучен ред. Текстове се извеждат без кавичките. Общият брой редове в изхода трябва да бъде равен на броя на нишките плюс броя на заключените мутекси.

Ограничения: $1 \le M \le 10$, $1 \le N_i \le 10$**.** Изпълнението на нишките продължава до достигането на състояние на пълен застой, при който нито една от нишките не може да продължи да се изпълнява понеже или е приключила, или е блокирана от deadlock.

Пример 1. Вход	**Изход**

`2`
`1`
`HLD X`
`2`
`HLD Y`
`HLD X`
	`Thread 1 completes successfully.`
	`Thread 2 waits on command 2 for mutex X.`
	`Mutex X is occupied by thread 1.`
	`Mutex Y is occupied by thread 2.`

Пример 2. Вход	**Изход**

`3`
`4`
`HLD A`
`HLD E`
`RLS A`
`RLS E`
`6`
`HLD E`
`HLD D`
`HLD A`
`RLS E`
`RLS D`
`RLS A`
`4`
`HLD X`
`RLS X`
`HLD X`
`RLS X`
	`Thread 1 waits on command 2 for mutex E.`
	`Thread 2 waits on command 3 for mutex A.`
	`Thread 3 completes successfully.`
	`Mutex A is occupied by thread 1.`
	`Mutex D is occupied by thread 2.`
	`Mutex E is occupied by thread 2.`