

КОНТРОЛНО НА НАЦИОНАЛНИЯ ОТБОР ЗА XVII МОИ

София, 19 Юни 2005

Задача 3. ФАБРИКА

Мечо Пух реши да си построи фабрика за мед. Инвестира (почти) всичкия мед, който има във фабриката и ето, че дойде големият ден, в който Пух трябва да я пусне в действие! Но се появи проблем – той все още не е наел работници... Пух има нужда от точно K на брой работника, за да функционира фабриката му – ни повече, ни по-малко. Замисли се Мечо кои от приятелите си да наеме, така че да не обиди никой от тях. Мисли, мисли, но нищо не измисли. Накрая изяде един буркан мед и му дойде наум следната гениална идея: ще състави списък на всички свои приятели и ще ги оцени по два показателя. Първият показател ще бъде възрастта, а вторият – теглото. Така например Прасчо е по-млад и по-лек от Йори, който пък е по-лек, но по-стар от Тигър. След това ще ги раздели на N групи по всеки от двата показателя. Групите и по двата показателя са номерирани с целите числа от 1 до N , като група i съдържа по-млади (или по-леки) животни от група j тогава и само тогава, когато $i < j$. Всеки приятел на Пух попада в точно една група по тегло и точно една група по възраст.

Така на всяко животно в гората Пух съпостави две числа – група по възраст и група по тегло. След това той ще се изхитри и ще постави в гората обява, че търси за фабриката си работници, които са едновременно не по-стари от някаква възрастова група и не по-тежки от някаква категория тегло. Така няма да обиди никой, а ще постигне целта си – ще наеме точно K от приятелите си.. Но как да разбере какви горни граници да обяви? Тук вече се появи проблем. За щастие Мечо Пух разбра, че в България точно по това време има състезание, на което се събират най-изобретателните умове в областта на информатиката, и реши да остави на тях да му помогнат да намери точните възраст и тегло, които да обяви в гората.

За нещастие списъкът с данните на животните е поверителен и Мечо не може да Ви го предостави, но може да ви дава отговор на следния въпрос: “Колко животни има в гората, които са едновременно в група по възраст с номер не по-голям от P и в група по тегло с номер не по-голям от Q ?”. Освен това, понеже времето е пари (а с всеки изминал ден Пух губи от меда, който фабриката можеше да произвежда), решението Ви трябва да отговори на Пух с колкото се може по-малко запитвания.

Напишете програма **factory**, която да помогне на Пух.

В началото на изпълнението си Вашата програма трябва да извика функцията

```
void init(long *K, long *N);  
procedure init(var K: LongInt; var N: LongInt);
```

която ще ви даде информация колко работника иска да наеме Пух, както и на колко групи е разделил Пух приятелите си по двата показателя. Вашата програма може да задава въпроси само за целите числа в интервала $[1;N], [1;N]$.

След това Вашата програма може да се обръща към библиотечната функция

```
long ask(long P,long Q);
function ask(P,Q : LongInt):LongInt;
```

Тази функция връща броят на приятелите, които са едновременно в група по възраст с номер не по-голям от P и в група по тегло с номер не по-голям от Q. В случай, че $P < 1$, $P > n$, $Q < 1$ или $Q > n$, извикването се счита за невалидно и програмата Ви получава 0 точки за съответния тест.

Когато определи отговора, решението Ви трябва да извиква функцията

```
void answer(long P,long Q);
procedure answer(P,Q: LongInt);
```

Числата P и Q трябва да са такива, че животните, отговарящи на условията, обявени от Пух, са точно K на брой. Гарантирано е, че такива две числа със сигурност има.

Ограничения:
 $1 \leq N \leq 1024$

Оценяване: Ако програмата Ви не извика функцията init() преди някоя от функциите ask() и answer(), извика функцията init() повече от веднъж, извика функцията ask() повече от веднъж за едни и същи P и Q, даде грешен отговор или направи невалидно извикване към функцията ask(), то тя ще получи 0 точки за съответния тест. В противен случай, ако програмата Ви намира верен отговор и прави не повече от M запитвания, тя ще получи пълния брой точки за съответния тест. Ако програмата ви направи повече от M запитвания, тя ще получи 0 точки. Числото M е функция на N (т.е. не зависи от останалите параметри на теста), такава че всеки тест с размер N може да бъде решен с не повече от M запитвания.

Пример:

init()	N=3 K=4	Има 3 групи по тегло и 3 групи по възраст. Пух иска да наеме 5 работника
ask(2,3)	7	Има 7 кандидата, които са едновременно в групи 1 или 2 по възраст и групи 1,2 или 3 по тегло
ask(1,1)	2	Кандидатите в група 1 по възраст и група 1 по тегло са двама
ask(1,3)	4	Кандидатите в група 1 по възраст и групи 1, 2 или 3 по тегло са общо трима.
answer(1,3)	-	Ако Пух определи за възрастова граница 1-ва група и граница за теглото 3-та група, то ще има точно K кандидата за работата.

За да тествате решението си, създайте файл **factory.in** в директорията, в която се намира модула. Файлът има следния формат: на първия ред съдържа две цели

числа F и N – съответно броят на приятелите на Пух и броят на групите. Всеки от следващите F реда трябва да съдържа по две цели числа всеки, описващи групата по възраст и групата по тегло на поредния приятел. Входният файл трябва да завършва с едно цяло число K – колко приятели иска да наеме Пух.

Модулът ще създаде изходен файл **factory.out** – на първият му ред ще е записан броят на заявките, които Вашето решение прави или -1 в случай на грешка. Вторият ред на файла ще съдържа текстова информация, описваща развитието на играта.

Пример:

factory.in

10 3

1 1

1 2

2 3

3 1

3 3

1 1

1 3

2 2

3 1

2 2

4