

**ШКОЛА ПО ИНФОРМАТИКА**  
**Смолян, 31 август – 5 септември 2019 г.**  
**7 клас**

**Задача 1. ОГРАДА**

Иван и Петко са селскостопански производители от село Стоичково. За да се справят с лошите метеорологични условия преди години, решили да си обединят нивите и да ги обработват съвместно. С появата на африканската чума по свинете и с цел по-сигурна защита Иван и Петко искат да се отделят един от друг и да построят висока ограда между нивите си. Оградата трябва да е права линия, която разделя нивите или от север на юг или от изток на запад. Има един единствен проблем – оградата не трябва да пресича 4-те навеса, които вече са изградени и след построяването ѝ на всеки трябва да се паднат по два навеса. За да определят мястото на оградата Иван и Петко взели карта на района, на която навесите са означени като правоъгълни области със страни, успоредни на хоризонталния и вертикален край на картата, и започнали да спорят къде е мястото на оградата.

Помогнете им за това като напишете програма **fence**, която да определи мястото на оградата.

**Вход**

От стандартния вход се въвеждат четири реда с по четири цели числа, разделени с по един интервал. Всеки ред задава координатите  $X$  - на долния ляв край и  $Y$  - горния десен край на един навес, определени спрямо координатна система, съвпадаща с долния хоризонтален и левия вертикален край на картата.

**Изход**

На първия ред на стандартния изход програмата трябва да изведе една от следните думи:

- impossible, ако не е възможно да се построи исканата ограда;
- horizontal, ако оградата трябва да е успоредна на долния хоризонтален край на картата;
- vertical, ако оградата трябва да е успоредна на левия вертикален край на картата.

Ако оградата може да се построи, на втория ред програмата трябва да изведе едно цяло число – съответната стойност на координатната ос, в която оградата ще я пресече.

**Ограничения**

$$0 \leq X, Y \leq 10^9$$

Навесите нямат общи точки.

Всеки навес има ненулева площ.

**Примери**

**Вход**

```
0 0 1 1
1 2 3 3
4 0 5 3
3 4 6 7
```

**Изход**

```
vertical
3
```

**Вход**

```
0 0 1 1
1 2 3 3
4 0 5 2
2 4 6 7
```

**Изход**

```
horizontal
2
```

**ШКОЛА ПО ИНФОРМАТИКА**  
**Смолян, 31 август – 5 септември 2019 г.**  
**7 клас**

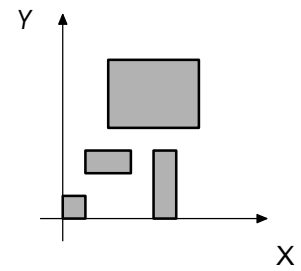
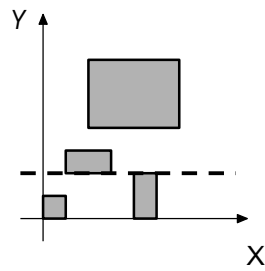
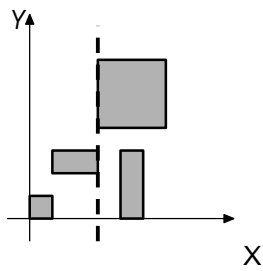
**Вход**

0 0 1 1  
1 2 3 3  
4 0 5 3  
2 4 6 7

**Изход**

impossible

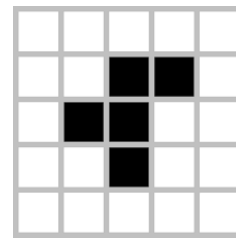
Графично представяне на трите примера:



**ШКОЛА ПО ИНФОРМАТИКА**  
**Смолян, 31 август – 5 септември 2019 г.**  
**7 клас**

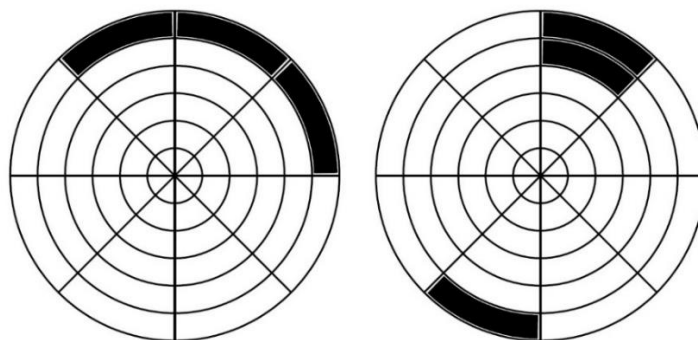
**Задача 2. ПОЛЯРЕН ЖИВОТ**

Играта “живот“, създадена от Джон Конвей, се развива в правоъгълна мрежа от клетки, някои от които са живи, а други са мъртви. Всички клетки едновременно актуализират състоянието си през равни интервали от време, като на всяка итерация образуват ново поколение клетки, използвайки следните правила:



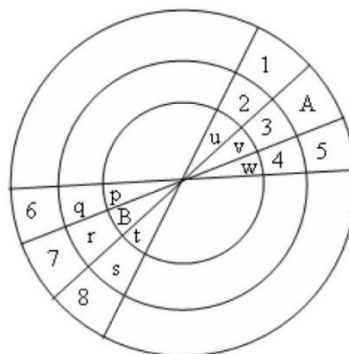
- в мъртва клетка с точно три живи съседа, на следващата итерация се ражда живот. Съседни на дадена клетка са осемте клетки, които я заобикалят;
- жива клетка с по-малко от два или повече от три живи съседа, умира в следващата итерация;
- състоянието на другите клетки остава непроменено.

Какво се променя обаче, ако заменим правоъгълната мрежа с „полярна“ мрежа като тази, показана на Фиг. 1:



Фиг. 1 Две итерации от играта върху „полярна“ мрежа.

Полярната мрежа се състои от концентрични окръжности и диаметри на най-външната окръжност. Диаметрите разделят големия кръг на еднакви сектори. Всяка клетка представлява сектор, затворен от две съседни окръжности и два съседни радиуса. Различават се само клиновидните клетки, образувани от най-вътрешната окръжност и радиусите. Всяка клетка има по осем съседни, с изключение на тези по външната обиколка и клиновидните клетки в центъра. За клетките от външния пръстен и клиновидните клетки в центъра, за съседни клетки ще считаме: петте обикновени съседни на клетката, плюс клетката, диаметрално противоположна на дадената клетка, плюс лявата и дясната съседна клетка на тази клетка. Погледни Фиг. 2:



Фиг. 2. Клетките, съседни на клетките от външния пръстен и клиновидните клетки в центъра.

**ШКОЛА ПО ИНФОРМАТИКА**  
**Смолян, 31 август – 5 септември 2019 г.**  
**7 клас**

Клетка **A** е от външния пръстен на мрежата, а клетка **B** е клиновидна от центъра на мрежата. Съседни на клетка **A** са клетките, номерирани от **1** до **8**, а съседни на **B** са клетките от **p** до **w**. Напишете програма **polar**, която реализира предварително зададен брой итерации на играта живот върху описаната по-горе „полярна” мрежа.

**Вход**

От първия ред на стандартния вход се въвеждат две положителни числа **M** и **N**, където **M** е броя на окръжностите, а **N** е броя на диаметрите. (на Фиг. 1 има 6 окръжности и 4 диаметъра). На втория ред е зададено цяло положително число **K**, което задава броя на живите клетки. От следващия ред се въвежда списък от **K** различни двойки положителни цели числа, които могат да са разположени на няколко реда. Всяка двойка определя местоположението на една жива клетка, като първото число указва номер на пръстен (пръстените са номерирани от външния пръстен навътре, като номерацията започва от 0), а второто число задава номера на клетката в този пръстен (номерацията започва от 0 и върви по часовниковата стрелка около пръстена, като започва от радиуса, който съвпада с положителната посока на оста Y.). От последния ред се въвежда неотрицателно цяло число **G**, което задава броя на итерациите.

**Изход**

На единствения ред на стандартния изход да се изведат пет цели числа, разделени с по един интервал: броя живи клетки след **G** итерации по правилата на играта “живот”; местоположението (**r1**, **c1**) на първата жива клетка и местоположението (**r2**, **c2**) на последната жива клетка, като живите клетки са подредени във възходящ ред по номерата на пръстените и за всеки пръстен, във възходящ ред по номерата на живите клетки. Ако няма останали живи клетки, се извежда: 0 -1 -1 -1 -1.

**Ограничения**

$$3 \leq M \leq 100$$

$$3 \leq N \leq 50$$

$$0 \leq G \leq 500$$

**Пример 1:**

**Вход:**

6 4  
3  
0 7 0 0  
0 1  
1

**Изход:**

3 0 0 1 0

**Пример 2:**

**Вход:**

4 3  
1  
1 0  
10

**Изход:**

0 -1 -1 -1 -1

**ШКОЛА ПО ИНФОРМАТИКА**  
**Смолян, 31 август – 5 септември 2019 г.**  
**7 клас**

**Задача 3. РАЗХОДКА**

Организаторите на школата по информатика решили да изненадат учениците с разходка до връх Снежанка. Това може да стане с лифт1, всяка седалка на който побира точно  $n_1$  на брой ученици, или с лифт2, всяка седалка на който превозва точно  $n_2$  на брой ученици.



Организаторите се чудят колко седалки от лифт1 и колко седалки от лифт2 да наемат така, че всичките  $n$  участници в школата да се съберат в тях, но също така - да наемат възможно най-малко на брой седалки.

Вие може да помогнете като напишете програма **walk**, която намира най-малкия брой седалки от двата вида, които организаторите трябва да наемат така, че всичките  $n$  ученици да се поберат в тях.

**Вход**

На първия ред на стандартния вход са записани три естествени числа:  $n$  – брой на учениците,  $n_1$  – брой на учениците, които могат да се съберат на една седалка на лифт1 и  $n_2$  - брой на учениците, които могат да се съберат на една седалка от лифт2. Трите числа са разделени с по един интервал.

**Изход**

На един ред на стандартния изход програмата трябва да изведе две цели числа – брой на седалките на лифт1 и брой на седалките на лифт2, така че общият брой седалки да е минимален.

Ако е невъзможно да се поберат всички ученици, програмата трябва да отпечатва “Impossible”.

**Ограничения**

$$1 \leq n_1, n_2 \leq 10^9$$

$$1 \leq n \leq 10^{18}$$

$$n_1 \neq n_2$$

**Пример 1**

**Вход**

43 3 4

**Изход**

1 10

**Пример 2**

**Вход**

40 9 12

**Изход**

Impossible