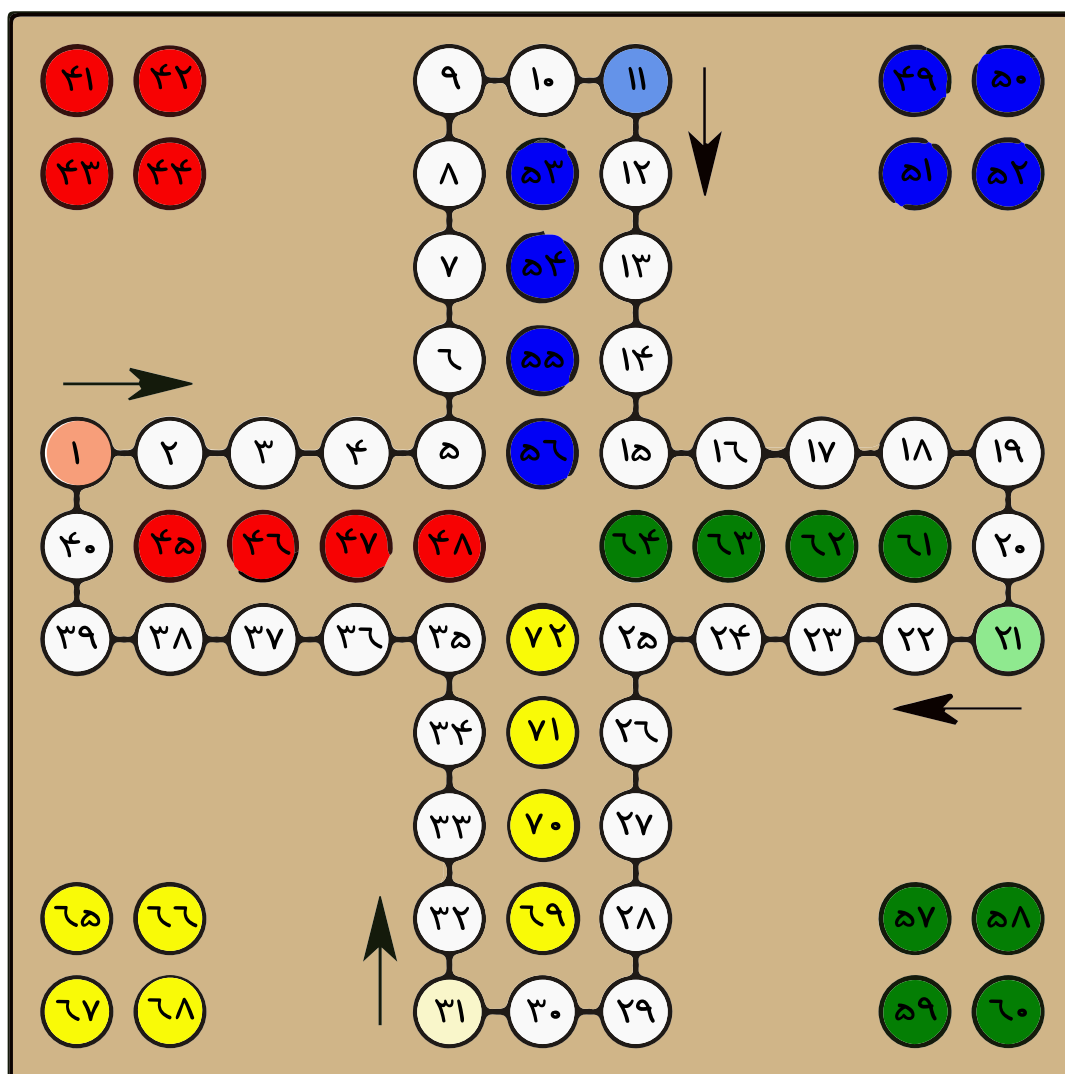


## آشنایی با منچ

در این قسمت می‌خواهیم به‌طور کلی مروری بر مقررات بازی کنیم.

منچ یک بازی حداکثر ۴ نفره است. هر کدام از این ۴ بازیکن یکی از رنگ‌های «قرمز»، «آبی»، «سبز» و «زرد» را انتخاب می‌کنند و ۴ مهره با رنگ خودشان دارند که در ۴ گوشه‌ی صفحه قرار دارند. (جمعاً ۱۶ مهره) هدف این است که هر بازیکن ۴ مهره رنگ خودش را به ۴ خانه هم‌رنگ خودش در وسط صفحه برساند.



بازی به نوبت انجام می‌شود، ابتدا نوبت بازیکن «قرمز»، سپس بازیکن «آبی»، بعد از آن بازیکن «سبز» و پس از آن نوبت بازیکن «زرد» است و همین ترتیب تکرار می‌شود.

هر کس در نوبت خودش تاس می‌اندازد. تاس عددی بین ۱ تا ۶ را نشان می‌دهد و می‌تواند مهره خودش را در جهت مشخص شده، به تعداد عدد ظاهر شده به جلو ببرد. هرکس برای اینکه یکی از مهره‌هایش را وارد نوار اصلی بازی کند، باید یک بار عدد ۶ بیاورد تا بتواند این کار را انجام دهد.

اگر عدد تاس بیشتر از تعداد خانه‌هایی باشد که پیش‌روی آن مهره قرار دارد یا در خانه‌ی مقصد یک مهره دیگر از همان بازیکن باشد، هیچ حرکتی انجام نمی‌شود. توجه کنید اگر مهره بازیکن دیگری باشد، مهره آن بازیکن به قسمت گوشه می‌رود. همچنین پریدن مهره‌ها از روی یک‌دیگر مجاز است.

از شما می‌خواهیم اتفاقات این بازی را شبیه‌سازی کنید.

## پروژه منچ: بخش ۱

- محدودیت زمان: ۱ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

از شما می‌خواهیم برنامه‌ای بنویسید که عدد ظاهر شده روی تاس را مانند ورودی‌های نمونه چاپ کند.

### ورودی

در تنها سطر ورودی، عدد صحیح و مثبت  $d$  آمده، که نشان‌دهنده‌ی عدد ظاهر شده‌ی تاس است.

$$1 \leq d \leq 6$$

### خروجی

مانند خروجی‌های نمونه، عدد ظاهر شده تاس را به صورت استاندارد چاپ کنید.

### مثال

ورودی نمونه ۱

1

خروجی نمونه ۱

.....  
 .....  
 ..0..  
 .....  
 .....

ورودی نمونه ۲

2

خروجی نمونه ۲

.....  
 ...0.  
 .....  
 .0...  
 .....

ورودی نمونه ۳

3

خروجی نمونه ۳

.....  
 ...0.  
 ..0..  
 .0...  
 .....

ورودی نمونه ۴

4

خروجی نمونه ۴

.....  
 .O.O.  
 .....  
 .O.O.  
 .....

ورودی نمونه ۵

5

خروجی نمونه ۵

.....  
 .O.O.  
 ..O..  
 .O.O.  
 .....

ورودی نمونه ۶

6

خروجی نمونه ۶

.....  
 .O.O.  
 .O.O.

.O.O.

.....

## پروژه منچ: بخش ۲

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

یکی از ابزارهایی که برای پیاده‌سازی بازی منچ به آن نیاز داریم، «تاس» است. به همین منظور، می‌خواهیم تابع `get_dice` را پیاده‌سازی کنیم. این تابع هیچ آرگومانی ورودی نمی‌گیرد. این تابع بعد از هر بار فراخوانی، مقدار عددی که روی تاس، بعد از انداختن ظاهر می‌شود را نشان می‌دهد.

برای رسیدن به این هدف، با داشتن سه عدد اول  $A, B$  و  $m$  دنباله تصادفی  $X$  را به این صورت می‌سازیم:

$$X_k = \begin{cases} B & k = 1 \\ (A.X_{k-1} + B) \bmod m & k > 1 \end{cases}$$

حال از روی دنباله‌ی  $X$  عدد ظاهر شده در پرتاب  $k$ ام را که با  $dice_k$  نشان می‌دهیم؛ از این رابطه بدست می‌آید:

$$dice_k = (X_k \bmod 6) + 1$$

از شما می‌خواهیم تابع `get_dice` را طوری پیاده‌سازی کنید که بعد از  $k$  بار صدا کردن مقدار  $dice_k$  را چاپ کند. یعنی در اولین فراخوانی `get_dice`، مقدار  $dice_1$ ، در دومین فراخوانی `get_dice`، مقدار  $dice_2$  و... برگردانده شود.

## ورودی

در سطر اول ورودی، به‌ترتیب سه عدد اول  $A, B$  و  $m$  داده می‌شود.

$$2 \leq A, B < m \leq 997$$

در سطر دوم ورودی، عدد صحیح و مثبت  $n$  داده می‌شود.

$$1 \leq n \leq 100\,000$$

## خروجی

خروجی  $n$  سطر دارد، عدد نوشته شده در سطر  $k$ ام مقدار تابع `get_dice` بعد از  $k$  بار فرخوانی است.

## مثال

### ورودی نمونه ۱

```
37 71 101
10
```

### خروجی نمونه ۱

```
6
1
3
5
4
3
4
4
3
4
```

### ورودی نمونه ۲

```
479 139 911
8
```



## خروجی نمونه ۲

2

2

1

2

5

4

6

4

### پروژه منچ: بخش ۳

- محدودیت زمان: ۱ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

در این بخش از پروژه، فرض کنید یک بازیکن داریم که تنهایی منچ بازی می‌کند. به‌طور دقیق‌تر، این بازیکن رنگ «قرمز» را انتخاب می‌کند؛ ۴ مهره قرمز، در ۴ خانه‌ی گوشه بالا سمت راست، قرار می‌دهد. این چهار مهره را با رشته‌های R1 ، R2 ، R3 و R4 نشان می‌دهیم.

این بازیکن  $q$  درخواست دارد. هر درخواست در یک سطر ورودی داده می‌شود.

#### پرتاب تاس

این درخواست یعنی بازیکن قرمز می‌خواهد تاس بی‌اندازد.

dice

این درخواست یعنی بازیکن قرمز، می‌خواهد تاس بیاندازد.

اگر عدد ظاهر شده از پرتاب‌های قبلی وجود دارد که هنوز حرکت متناسبی برای آن انجام نشده، پیام `invalid dice rolling` را در یک سطر چاپ کنید.

در غیر این صورت با استفاده از تابع `get_dice` که در بخش قبلی پیاده‌سازی شده، یک عدد دریافت کنید. اگر عدد ۶ ظاهر شد، این عدد را جزو اعداد ظاهر شده در نظر بگیرید و مجدداً تاس بیاندازید. این کار را آنقدر تکرار می‌کنیم که دیگر ۶ ظاهر نشود.

تضمین می‌شود که فرآیند پرتاب تاس، پایان پذیر باشد. از شما می‌خواهیم تمام اعداد ظاهر شده را در یک سطر، با یک فاصله از هم، چاپ کنید.

برای بهتر متوجه شدن این درخواست، مثال‌های نمونه را بررسی کنید.

## ورود مهره

این درخواست یعنی بازیکن قرمز، می‌خواهد مهره‌ی  $m$  را وارد بازی کند.

enter <m>

تضمین می‌شود  $m$  یکی از مقدارهای  $R1$ ،  $R2$ ،  $R3$  یا  $R4$  باشد.

- اگر مهره‌ی  $m$ ، در این لحظه در جریان بازی باشد؛ پیام `that is in` را در یک سطر چاپ کنید.
- اگر در پرتاب‌های قبلی تاس، عدد ۶ ظاهر نشده باشد تا آن را برای ورود مصرف کنیم، پیام `you need six` را در یک سطر چاپ کنید.
- اگر خانه‌ی اولیه برای شروع با مهره‌ی قرمز دیگری اشغال شده باشد، پیام `busy starting cell` را در یک سطر چاپ کنید.

اگر چند مورد از شرایط بالا برقرار بود، موردی که زودتر بیان شد را در نظر بگیرید و پیام آن را چاپ کنید.

در صورتی که هیچ‌کدام از حالت‌های بالا اتفاق نیافتد، یکی از ۶‌های ظاهر شده در پرتاب تاس را حذف کنید و مهره‌ی  $m$  را وارد خانه اولیه قرمز (خانه ۱) کنید و شماره‌ی خانه‌ای که مهره وارد آن شده را در یک سطر چاپ کنید.

برای بهتر متوجه شدن این درخواست، مثال‌های نمونه را بررسی کنید.

## حرکت مهره

این درخواست یعنی بازیکن قرمز می‌خواهد مهره‌های  $m$  خود را  $s$  واحد حرکت دهد.

move <m> <s>

تضمین می‌شود  $m$  یکی از مقدارهای  $R1$ ،  $R2$ ،  $R3$  یا  $R4$  باشد. همچنین تضمین می‌شود  $s$  یکی از اعداد 1 تا 6 باشد.

- این درخواست زمانی معتبر است که عدد  $s$  در یکی از پرتاب‌های قبلی تاس، ظاهر شده باشد ولی از

آن استفاده نشده باشد. اگر این درخواست معتبر نبود، پیام `invalid move` را در یک سطر چاپ کنید.

- اگر مهره  $m$  در چهار خانه‌ی گوشه جدول است، پیام `it is not in` را در یک سطر چاپ کنید.
- اگر  $s$  خانه بعدی آن وجود نداشته باشد. (به خانه‌های پایانی نزدیک باشیم.) پیام `you can not move` را در یک سطر چاپ کنید.
- اگر خانه‌ی مقصد، توسط مهره‌ی قرمز دیگری اشغال شده است، پیام `destination is busy` را در یک سطر چاپ کنید.

اگر چند مورد از شرایط بالا برقرار بود، موردی که زودتر بیان شد را در نظر بگیرید و پیام آن را چاپ کنید. در صورتی که هیچ‌کدام از حالت‌های بالا اتفاق نیافتد، یکی از  $s$ های ظاهر شده در پرتاب تاس را حذف کنید و مهره  $m$  را  $s$  واحد به جلو ببرید و شماره‌ی خانه‌ای که مهره وارد آن شده را چاپ کنید. برای بهتر متوجه شدن این درخواست، مثال‌های نمونه را بررسی کنید.

## رها کردن

این درخواست یعنی بازیکن می‌خواهد از همه‌ی اعداد ظاهر شده توسط تاس صرف نظر کند و هیچ حرکتی به‌ازای این اعداد انجام ندهد.

`giveup`

این درخواست باید باعث لغو شدن همه‌ی پرتاب‌های تاسی شود که تا کنون انجام شده است.

برای بهتر متوجه شدن این درخواست، مثال‌های نمونه را بررسی کنید.

## ورودی

در سطر اول ورودی، به‌ترتیب سه عدد اول  $A$ ،  $B$  و  $m$  داده می‌شود.

$$2 \leq A, B < m \leq 997$$

در سطر دوم ورودی، عدد صحیح و مثبت  $q$  داده می‌شود.

$$1 \leq q \leq 100\,000$$

در  $q$  سطر بعدی، در هر سطر، یکی از چهار درخواست که توضیح آن در متن سوال آمده می‌آید.

## خروجی

خروجی متناسب با هر درخواست را در یک سطر جداگانه چاپ کنید.

## مثال

### ورودی نمونه ۱

```
37 71 101
14
dice
enter R2
move R2 1
dice
move R2 3
dice
move R2 5
dice
move R2 4
dice
giveup
move R2 3
dice
move R2 4
```

### خروجی نمونه ۱

```
6 1
1
```

2  
3  
5  
5  
10  
4  
14  
3  
invalid move  
4  
18

## ورودی نمونه ۲

479 139 911  
8  
move R3 2  
enter R2  
dice  
dice  
enter R4  
giveup  
enter R4  
dice

## خروجی نمونه ۲

invalid move  
you need six  
2  
invalid dice rolling  
you need six  
you need six  
2

## پروژه منچ: بخش ۴

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در این بخش می‌خواهیم بازی را برای چهار بازیکن پیاده‌سازی کنیم.

مهره‌های بازیکن «قرمز» R1 ، R2 ، R3 و R4 هستند که در ابتدا به ترتیب در خانه‌های 41، 42، 43 و 44 قرار دارند. خانه‌ی شروع برای «قرمز» 1 است. هدف بازیکن قرمز این است که این ۴ مهره را (به هر ترتیبی) به خانه‌های 45، 46، 47 و 48 ببرد.

مهره‌های بازیکن «آبی» B1 ، B2 ، B3 و B4 هستند که در ابتدا به ترتیب در خانه‌های 49، 50، 51 و 52 قرار دارند. خانه‌ی شروع برای «آبی» 11 است. هدف بازیکن قرمز این است که این ۴ مهره را (به هر ترتیبی) به خانه‌های 53، 54، 55 و 56 ببرد.

مهره‌های بازیکن «سبز» G1 ، G2 ، G3 و G4 هستند که در ابتدا به ترتیب در خانه‌های 57، 58، 59 و 60 قرار دارند. خانه‌ی شروع برای «سبز» 21 است. هدف بازیکن قرمز این است که این ۴ مهره را (به هر ترتیبی) به خانه‌های 61، 62، 63 و 64 ببرد.

مهره‌های بازیکن «زرد» Y1 ، Y2 ، Y3 و Y4 هستند که در ابتدا به ترتیب در خانه‌های 65، 66، 67 و 68 قرار دارند. خانه‌ی شروع برای «قرمز» 31 است. هدف بازیکن قرمز این است که این ۴ مهره را (به هر ترتیبی) به خانه‌های 69، 70، 71 و 72 ببرد.

نکاتی که باید در این حالت در نظر بگیریم.

### قوانین نوبت

نوبت‌ها به ترتیب برای رنگ‌های «قرمز»، «آبی»، «سبز» و «زرد» است و بعد از یک دور گردش نوبت‌ها، مجدداً به همین ترتیب نوبت آن‌ها می‌شود. این روند تا آخر ادامه دارد.

هر بازیکن در نوبت خودش، باید یکبار تاس را با موفقیت بریزد. (توجه کنید اگر ۶ ظاهر شود به صورت خودکار باید روند تاس ریختن ادامه یابد.)

سپس باید با سه درخواست «ورود مهره»، «حرکت مهره» و «رها کردن» **همه‌ی** حرکات تاسی که انداخته را تمام کند.

توجه کنید زمانی که کار با ارسال پیام تمام شود، یعنی عملیات با موفقیت انجام نشده، پس نوبت‌ها نیز تغییری نمی‌کنند.

اگر نوبت یک رنگ باشد هر درخواستی که بخواهد باعث تغییر در وضعیت مهره‌های دیگر ایجاد کند باید با پیام `it is not your turn` را چاپ کنید.

## قوانین حذف

اکنون که چند مهره وجود دارد، ممکن است حرکت یک مهره به خانه‌ای باشد که مهره‌ای از رنگ دیگر وجود دارد. در این صورت شما باید مهره‌ای که در آن خانه قرار دارد را از آن نقطه بردارید و به اولین (کم شماره ترین) خانه‌ی گوشه‌ای مربوط به آن رنگ ببرید.

## درخواست‌ها

در خواست‌ها همان حالت‌های بخش قبلی را دارند ولی حالت‌های مهره ( $m$ ) می‌تواند هر کدام از ۱۶ رشته‌ی بالا باشد.

## ورودی

در سطر اول ورودی، به ترتیب سه عدد اول  $A$ ،  $B$  و  $m$  داده می‌شود.

$$2 \leq A, B < m \leq 997$$

در سطر دوم ورودی، عدد صحیح و مثبت  $q$  داده می‌شود.

$$1 \leq q \leq 1000$$



در  $q$  سطر بعدی، در هر سطر، یکی از چهار درخواست که توضیح آن در متن سوال آمده می‌آید.

## خروجی

خروجی متناسب با هر درخواست را در یک سطر جداگانه چاپ کنید.

## مثال

### ورودی نمونه ۱

```
41 41 997
7
dice
enter R2
move R2 6
move R2 2
dice
enter B1
move B1 3
```

### خروجی نمونه ۱

```
6 6 2
1
7
9
6 3
11
14
```

## پروژه منچ: بخش ۵

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در این بخش از پروژه، می‌خواهیم بعد از هر عملیات، چه موفق چه ناموفق وضعیت صفحه‌ی بازی نمایش داده شود.

صفحه بازی بدون مهره‌ها به صورت زیر است.

```
XX.XX.....00.00.00.....XX.XX
XX.XX.....00.XX.00.....XX.XX
.....00.XX.00.....
.....00.XX.00.....
00.00.00.00.00.XX.00.00.00.00.00
00.XX.XX.XX.XX...XX.XX.XX.XX.00
00.00.00.00.00.XX.00.00.00.00.00
.....00.XX.00.....
.....00.XX.00.....
XX.XX.....00.XX.00.....XX.XX
XX.XX.....00.00.00.....XX.XX
```

هر دو خانه‌ی 00 و xx یکی از خانه‌ها را نشان می‌دهد و . خالی بودن صفحه را نشان می‌دهد.

حال از شما می‌خواهیم وضعیت مهره‌ها در صفحه را به این صورت نمایش دهید.

## مثال

### ورودی نمونه ۱

41 41 997

7

dice

```

enter R2
move R2 6
move R2 2
dice
enter B1
move B1 3

```

## خروجی نمونه ۱

```

6 6 2
R1.R2.....00.00.00.....B1.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.00.....
00.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...XX.XX.XX.XX.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
1
R1.xx.....00.00.00.....B1.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.00.....
R2.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...XX.XX.XX.XX.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
7
R1.xx.....00.00.00.....B1.B2
R3.R4.....00.xx.00.....B3.B4
.....R2.xx.00.....
.....00.xx.00.....

```

```

00.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...xx.xx.xx.xx.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
9
R1.xx.....R2.00.00.....B1.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.00.....
00.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...xx.xx.xx.xx.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
6 3
R1.xx.....R2.00.00.....B1.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.00.....
00.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...xx.xx.xx.xx.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
11
R1.xx.....R2.00.B1.....xx.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.00.....
00.00.00.00.00.xx.00.00.00.00.00
00.xx.xx.xx.xx...xx.xx.xx.xx.00
00.00.00.00.00.xx.00.00.00.00.00
.....00.xx.00.....

```

```

.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4
14
R1.xx.....R2.00.00.....xx.B2
R3.R4.....00.xx.00.....B3.B4
.....00.xx.00.....
.....00.xx.B1.....
00.00.00.00.00.00.xx.00.00.00.00
00.xx.xx.xx.xx...xx.xx.xx.xx.00
00.00.00.00.00.xx.00.00.00.00
.....00.xx.00.....
.....00.xx.00.....
Y1.Y2.....00.xx.00.....G1.G2
Y3.Y4.....00.00.00.....G3.G4

```