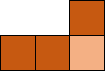
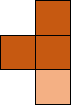
Росен се записа на състезание по древната игра „Блокчета“. Тя се играе на таблица с размер N\*N. На всеки ход играчът получава едно от 5-те възможни блокчета:

* Блокче 0: 
* Блокче 1: 
* Блокче 2: 
* Блокче 3: 
* Блокче 4: 

Всяко блокче се състои от няколко квадратчета, които заемат точно една клетка във таблицата. С по-светлия цвят е отбелязано знаковото квадратче, което се използва за отбелязване на позициите при ходовете (това ще обясним подробно след малко). Освен това блокчето, което получават, може да е завъртяно 0, 90, 180 или 270 градуса по часовниковата стрелка (знаковото квадратче също се върти). Целта на играча е да постави блокчето на определено място в таблицата, така че всички позиции, върху които се поставя блокчето, да са свободни. Ако 1) блокчето не може да се постави никъде на свободни позиции; 2) играчът доброволно се откаже или 3) времето изтече, играта приключва и резултатът на играча е броят ходове до момента.

Тъй като играчите постоянно запълват квадратчетата на таблицата с блокчета, има и специално правило, което да освобождава позиции. В края на всеки ход, комисията проверява всеки ред и всяка колона на таблицата. Ако някой ред или колона е напълно запълнен, всички квадратчета в него се изтриват. Проверката се прави едновременно за всички редове и колони, т.е. възможно е една клетка в таблицата да бъде изчистена и от реда си, и от колоната си.

Тъй като играта има дълга история, в началото ѝ не са съществували компютри, които да генерират произволни блокчета и произволни ротации. Затова комисията е измислила начин да си генерират тези данни сами. Тази информация е напълно секретна, но Росен успя да се сдобие с нея. Комисията избира 6 числа: a, b, c, d, e, f. Преди всеки ход се изпълняват следните операции:

* c = (c XOR a) + b
* f = (f XOR d) + e

След което блокчето, което получава играчът на този ход, е това с номер c mod 5. А числото f mod 4 индикира ротацията на блокчето: тя е 90 \* (f mod 4) градуса.

Росен много иска да спечели състезанието, но в момента всички участници са на ритуал по пречистване. Преди началото Росен успя да ви каже N, a, b, c, d, e и f и вашата работа е да напишете програма, която връща възможно най-дълга последователност от валидни ходове.

**Вход**

От първия ред на файла **block.in** се въвеждат N, a, b, c, d, e и f – големината на таблицата и шестте числа, определящи произволните блокчета и ротациите.

**Изход**

На първият ред във файла **block.out** отпечатайте числото M: броя ходове, които сте намерили. Всеки от следващите M реда трябва да съдържа две числа xi,yi: координатите на знаковото квадратче. Редовете и колоните са номерирани от 1 до N, в посока отляво-надясно и отгоре-надолу.

**Оценяване**

Ако 1) поставите блокче, като част от него излиза от таблицата ; 2) поставяте блокче на позиция, в която има запълнено квадратче, ще получите съобщение “Error” и 0 точки за съответния тест. В противен случай:

Нека maxScore е най-дългата последователност от ходове измежду всички участници. Ще получите умножено по точките, предвидени за този тест.

**Ограничения**

**Ограничение по време: 5 sec.**

**Ограничение по памет: 256 MB.**

**Разпределение на тестовете:**

|  |  |
| --- | --- |
| N | Процент от тестовете |
| N = 8 | 25% |
| N = 16 | 25% |
| N = 20 | 25% |
| N = 25 | 25% |

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (block.in)** | **Изход (block.out)** |
| 8 33 11 17 77 26 10 | 9  7 1  7 4  2 7  3 6  5 7  6 5  8 7  1 1  4 7 |

**Обяснение на примера**

За всеки ход е показана дъската след поставянето на блокче от Росен и след изчистването от комисията:

1. =>
2. =>
3. =>
4. =>
5. =>
6. =>
7. =>
8. =>
9. =>

Росен извършва 9 хода, след което се отказва. Резултатът му е 9.