

Mistrzostwa Polski Szkół Średnich w Programowaniu Zespołowym

Krzyżyki (K)

Limit pamięci: 1024 MB

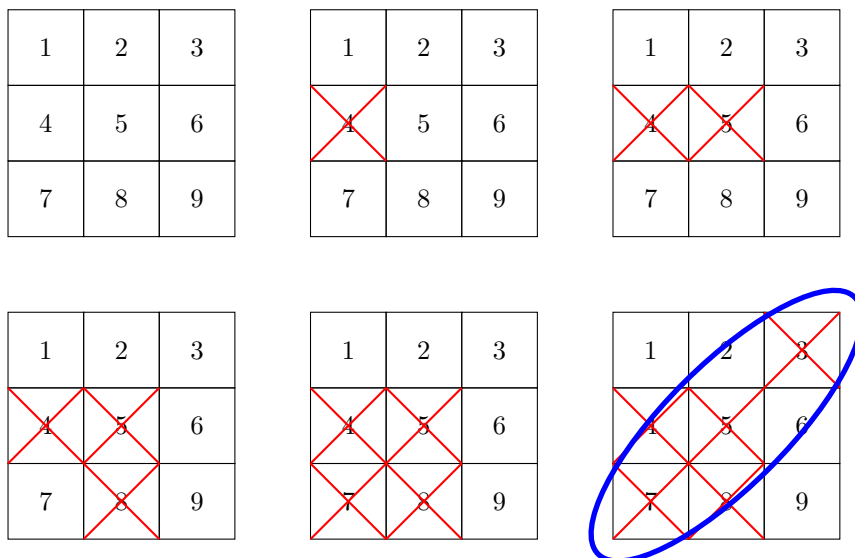
Limit czasu: 5.00 s

To zadanie jest interaktywne. Po wypisaniu każdego wiersza należy opróżnić bufor. W C++ możesz użyć `cout << flush`, w Pythonie – `sys.stdout.flush()`. Pamiętaj o wypisaniu białego znaku (na przykład końca linii) podczas opróżniania bufora. Należy ściśle trzymać się protokołu opisanego w rozdziale *Protokół interakcji* – niewysłanie wiadomości oczekiwanej przez interaktor może skutkować werdyktem *time limit exceeded* lub innym.

Bycie żółciem morskim wiąże się z poważnymi niedogodnieniami. Dla przykładu, weźmy grę w kółko i krzyżyk. Żółcie nie potrafią pisać, więc muszą sobie z nią radzić jakoś inaczej. Zaznaczanie krzyżyków jest proste – w morzu pływa masa plastikowych słomek, idealnie nadających się do tego celu. Z kółkiem jest nieco trudniej, okrągły kształt jest ciężki do odwzorowania. Należy zatem odpowiednio zmodyfikować zasady: gracze na zmianę stawiają krzyżyki na planszy 3×3 . Wygrywa ten, kto pierwszy zamknie jakąś linię (wiersz, kolumnę lub przekątną). Gra ta ma bardzo proste zasady, a mimo to sprawiała żółciom przyjemność.

Jednak w pewnym momencie, gra przestała być taka jak wcześniej, ponieważ zauważyłeś, że gracz rozpoczynający ma przewagę i jest w stanie zawsze wygrać. Naturalnie, stwierdzenie to nie zostało dobrze odebrane i większość żółwi posądziło Cię o kłamstwo. By bronić honoru, uznałeś, że zagrasz z nimi tyle gier ile tylko zechcą i wszystkie z nich wygrasz. Jeśli Ci się nie uda, zobowiązałeś się zjeść część słomek. Jak wiadomo, jedzenie słomek jest niezdrowe, dlatego lepiej tego unikać.¹

Ilustracja drugiej gry z przykładowej interakcji:



¹Byłoby to dobre miejsce, żeby przestrzec przed jedzeniem plastikowych słomek. Jednak w ostatnich czasach łatwiej znaleźć żółwia niż plastikową słomkę, więc raczej ostrzeżenie to byłoby zbędne. Żółwi też nie jedzą – mają w sobie za dużo plastiku. Gdybyś się jednak zdecydował, przepis na zupę z żółwia możesz odebrać u Jury.

Protokół interakcji

Na początku należy wczytać liczbę całkowitą T , będącą liczbą gier. Następnie T razy wykonywane jest poniższe:

Rozpoczyna się gra. W każdej rundzie Twój program powinien wypisać numer pola – cyfrę od 1 do 9 (tak jak na rysunku). Jeśli powstała linia 3 krzyżyków, gra się kończy i program powinien przejść do kolejnej gry. W przeciwnym przypadku, Twój program powinien wczytać numer pola, na którym ruch robi przeciwnik. Jeśli po tym ruchu powstała linia krzyżyków, Twój program przegrał. W przeciwnym wypadku zostanie rozegrana kolejna runda gry.

Jeśli Twój program przegrał w danej grze, powinien od razu skończyć działanie, aby dostać werdykt WA. Inaczej, możesz otrzymać werdykt TLE lub inny niesprecyzowany.

Po wygraniu T -tej gry program powinien zakończyć działanie.

Ograniczenia

$1 \leq T \leq 1000$.

Przykładowa interakcja

Wejście **Wyjście**

2

1

2

3

4

5

8

7

3

W pierwszej grze ruchy wykonywane są kolejno na polach 1, 2, 3. Te trzy pola tworzą linię. W drugiej grze ruchy wykonywane są kolejno na polach 4, 5, 8, 7, 3. Pola 3, 5, 7 tworzą linię

Narzędzia do testowania lokalnego

W plikach umieszczony został program **play.py**. Skrypt ten pozwala grać ze swoim programem. Można go uruchomić za pomocą komendy:

- dla plików w C++:

```
python3 play.py ./program
```

gdzie `./program` jest plikiem wykonywalnym z Twoim rozwiązaniem,

- dla plików w Pythonie:

```
python3 play.py program.py
```