Treść zadania i garść wskazówek podpowiadających na co będziemy patrzeć - poniżej. Ważne dla nas jest również to, jak piszesz kod, więc nawet jeśli nie wszystko się uda – nie poddawaj się i nie wyrzucaj nic do kosza! Nawet niedziałający kod może pokazać, że masz potencjał na dobrego programistę/dobrą programistkę.

Zasady zadania:

- Język wykorzystany do napisania programu to Java, PHP, C# lub TypeScript.
- Limit czasu wykonywania programu wynosi 30 sekund.
- Nie ma znaczenia czy program wykona się szybciej, czy wolniej; byle nie dłużej niż 30 sekund.
- W rozwiązaniu program ma wypisać dokładnie i wyłącznie to, co wskazane.
- Wielkość liter w danych wejściowych i wynikach jest istotna.
- Program powinien być odporny na błędne dane wejściowe i w przypadku błędnych danych wejściowych, program powinien wypisać wyłącznie słowo "error" (bez cudzysłowu).
- Kod powinien być możliwie czytelny.

Opis zadania
Należy przygotować symulację poruszania się figur po planszy. Figury należą do dwóch drużyn A i B i startują po przeciwległych stronach planszy, wypełniając wszystkie miejsca w rzędzie – podobnie do figur w grze w szachy.

Każda figura startuje z wyznaczonej pozycji x, y i charakteryzuje się stałą prędkością teleportacyjną v, która oznacza o ile komórek przeteleportuje się figura w osi Y w każdej turze symulacji. Tury są atomowe, tj. w jednej chwili wszystkie figury znikają i chwilę później (również jednocześnie) - pojawiają się na nowych miejscach.

Jeśli dwie figury pojawią się w tym samym miejscu, figura o mniejszej wartości bezwzględnej prędkości jest likwidowana. W przypadku remisu, usuwane są obydwie figury. Usuwane są również figury opuszczające planszę.

Symulacja kończy się w momencie kiedy jedna z drużyn nie posiada już żadnych figur.

Sposób określania parametrów
Dany jest strumień wejściowy – plik tekstowy. Prawidłowy strumień wejściowy zawiera dokładnie 6 wierszy w następującym formacie:

```
<nazwa_drużyny_A>
<współczynnik_prędkości_drużyny_A>
<nazwa_drużyny_B>
<współczynnik_prędkości_drużyny_B>
<wielkość_planszy_x>
<wielkość_planszy_y>
```

Gdzie:

- <nazwa_drużyny_A> oraz <nazwa_drużyny_B> to nazwy alfanumeryczne (A-Za-z0-9), niepuste, nie dłuższe niż 10 znaków, nie mogą być identyczne
- <współczynnik_prędkości_drużyny_A> to liczba całkowita od 1 do 3
- <współczynnik_prędkości_drużyny_B> to liczba całkowita od 1 do 3
- <wielkość_planszy_x> to liczba całkowita od 1 do 1000

- <wielkość_planszy_y> to liczba całkowita od 1 do 1000

Przykłady niepoprawnych nazw drużyn:

- SweetVictory123
- Team.*
- DrużynaĄ

Przykłady poprawnych nazw drużyn:

- BestTeam
- team2

Program nie przyjmuje dodatkowych parametrów. Prędkość poszczególnych figur obliczana jest wzorem:

- Dla figur o parzystej współrzędnej x
  v = 1 * <współczynnik_prędkości_drużyny>
- Dla figur o nieparzystej współrzędnej x
  v = 2 ^ <współczynnik_prędkości_drużyny>
- W przypadku figur zaczynających u góry planszy, należy prędkość przemnożyć przez -1 (aby poruszały się one w stronę figur zaczynających u dołu)

Po zakończeniu symulacji, program powinien wypisać nazwę drużyny która posiada nadal figury, jeżeli taka istnieje.
Jeśli na planszy nie zostały żadne figury, program powinien wypisać słowo „remis” (bez cudzysłowu).

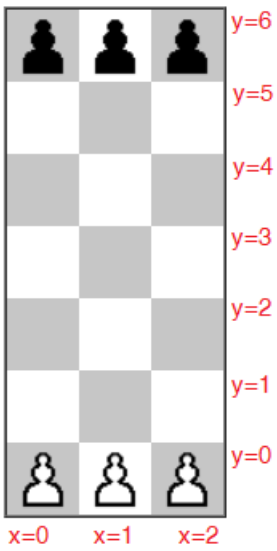
Przykładowy wynik działania programu
(przedstawione graficznie wizualizacje mają za zadania jedynie ułatwić zrozumienie i nie stanowią rozwiązania zadania):

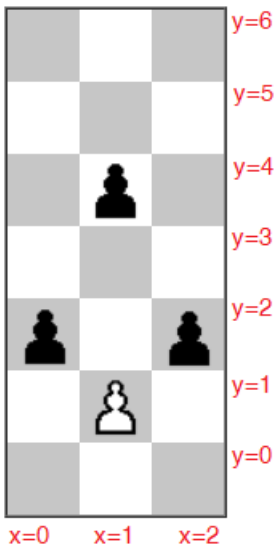
- dla przykładowego strumienia wejściowego (w formie pliku tekstowego)

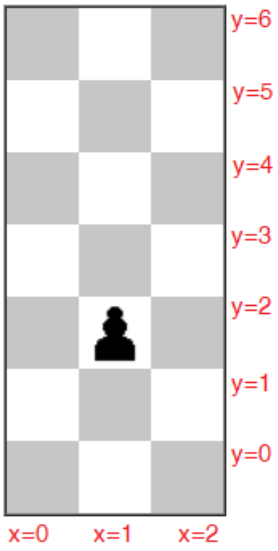
```
Tokyo
1
Berlin
2
3
7
```

wynikiem uruchomienia programu będzie:
```
Tokyo
```

Poniżej zamieszczono wizualizacje przedstawiające układ figur na planszy w momencie rozpoczęcia symulacji i po dwóch kolejnych turach. W każdej turze, figury poruszają się z wyliczoną dla nich prędkością i znikają zgodnie z zasadami eliminacji.







W ostatniej turze jedyna figura jaka została na planszy należy do drużyny Tokyo – drużyna ta wygrywa.

Plik z programem powinien nazywać się Main i posiadać rozszerzenie odpowiednie dla wybranego języka programowania (np. Main.java).

Podpowiedzi jak uzyskać dużo punktów (tj. więcej niż 0):

- Przed wysłaniem sprawdź, czy program działa.
- A potem sprawdź jeszcze raz (ale wciąż PRZED wysłaniem) czy program działa, przynajmniej dla danych wejściowych z treści zadania (inne testy, to już Twoja sprawa)
- A potem sprawdź jeszcze raz (ale wciąż PRZED wysłaniem) czy program działa.

-------- UWAGA --------

Kod programu opublikuj w swoim prywatnym repozytorium na GitHub i udostępnij użytkownikowi `biznesport-pl` przez zaproszenie (Invite collaborator).

Po wysłaniu zaproszenia do współpracy nie zapomnij dać nam znać mailem z jakiego konta na GitHub udostępniasz nam swoje rozwiązanie (wiadomość z GitHub podaje tylko Twój nick tam i nie zawsze jesteśmy w stanie połączyć go z Twoim adresem email).

Powodzenia!

Coś jest niejasne, masz jakieś pytania, chcesz wiedzieć więcej?
Po prostu odpisz na tę wiadomość – jesteśmy po drugiej stronie!

Pozdrawiamy
BiznesPort.pl