

Wydział

PREOI 2025

Dzień 6 – 30 stycznia 2025

Kod zadania: **wyd**
Limit pamięci: **256 MiB**



Jako młody, głupi i ambitny student, Bajtazar postanowił zapisać się na wiele przedmiotów, oczywiście wiele z nich w tym samym czasie. Zupełnie nie odnajduje się on w budynku wydziału i nigdy nie wie gdzie powinien się kierować. Widząc to, jego kolega Bitek postanowił mu pomóc.

Plan wydziału jest kwadratową siatką sal o wymiarach $n \times n$. Sala w lewym górnym rogu ma współrzędne $(0, 0)$, w prawym dolnym $(n - 1, n - 1)$, a w lewym dolnym $(n - 1, 0)$.

Bajtazar zapisał się na $K = 7$ przedmiotów, z których każdy prowadzony jest w **innej** sali. Co więcej, żaden przedmiot nie jest prowadzony w sali na brzegu wydziału. Brzegiem wydziału nazywamy sale w: zerowym wierszu, zerowej kolumnie, $n - 1$ -szym wierszu i $n - 1$ -szej kolumnie.

W każdej sali Bitek napisze jedną liczbę całkowitą z przedziału $[1, 10^9]$. Na podstawie tych liczb Bajtazar będzie kierował się w stronę właściwej sali. Jako że dewastowanie wydziału nie jest mile widziane przez Dziekana, Bitek chciałby minimalizować pisane przez siebie liczby.

Bajtazar pojawia się w jakiejś sali (również nie na brzegu). Jedyną informacją, jaką posiada, są liczby wpisane przez Bitka w 9 najbliższych salach. Na tej podstawie może podjąć jedną z pięciu akcji:

- 0 – przejście w prawo
- 1 – przejście w lewo
- 2 – przejście w dół
- 3 – przejście w górę
- 4 – Bajtazar myśli, że jest w dobrej sali i zostaje w miejscu

Bajtazar chciałby, żeby wykonana przez niego akcja była przejściem w stronę poszukiwanej sali (czyli musi pójść do takiego pola, w którym minimalna odległość do poszukiwanego pola jest jak najmniejsza). W przypadku, gdy takich akcji jest kilka, może wybrać **dowolną** z nich. Co więcej, Bajtazar powinien zwrócić właściwą wartość dla wszystkich 7 przedmiotów naraz (dla tych samych numerów wpisanych w pola przez Bitka).

Twoim zadaniem jest zaimplementowanie strategii dla Bitka i Bajtazara.

Biblioteka

Jest to zadanie interaktywne, to znaczy Twój program będzie porozumiewał się z biblioteką. Aby użyć biblioteki, należy załączyć nagłówek `#include "wydlib.h"`.

Wymagane funkcje

Twoim zadaniem jest zaimplementowanie dwóch funkcji:

- `std::vector<std::vector<int>> > Bitek(int n, std::array<int, K> A, std::array<int, K> B)`
Jest to funkcja, która powinna reprezentować strategię Bitka. Parametry n ($5 \leq n \leq 100$) oznacza wymiary wydziału.



Tablice A i B reprezentują współrzędne sal, w których prowadzone są zajęcia. Dla każdego i ($0 \leq i < K$) sala numer i ma współrzędne $(A[i], B[i])$. Zachodzi ($1 \leq A[i], B[i] < n - 1$). Funkcja ta powinna zwracać wektor wektorów $n \times n$, reprezentujący wartości zapisane przez Bitka w poszczególnych salach.

- `std::array<int, K> Bajtazar(std::array<int, 9> A)`
Jest to funkcja, która powinna reprezentować strategię Bajtazara.
Założmy, że Bajtazar znajduje się w sali (a, b) , ($1 \leq a, b < n - 1$). Wówczas tablica A zawiera wartości znajdujące się odpowiednio w salach:

$(a - 1, b - 1), (a - 1, b), (a - 1, b + 1), (a, b - 1), (a, b), (a, b + 1), (a + 1, b - 1), (a + 1, b), (a + 1, b + 1)$

Funkcja ta powinna zwrócić tablice numerów akcji podjętym przez Bajtazara dla każdego przedmiotu. Dokładniej, na i -tej pozycji powinna być akcja podjęta przez Bajtazara w celu dotarcia na zajęcia i po jak najkrótszej drodze.

Możesz tworzyć dodatkowe funkcje pomocnicze, ale **nie implementuj** funkcji `main`. Dodatkowo funkcje `Bitek` i `Bajtazar` nie powinny się komunikować. Pojedynczy test składa się z q ($1 \leq q \leq 300$) przypadków testowych, każdy polega na jednym wywołaniu funkcji `Bitek` i jednym wywołaniu funkcji `Bajtazar`.

Błędy

Zarówno biblioteka, z której korzysta program sprawdzający, jak i ta udostępniona w zakładce "Pliki", zwracają określone błędy przy niepoprawnym korzystaniu z biblioteki, razem z werdyktem *Zła Odpowiedź*. Są to:

- `WRONG_ANSWER 1` - funkcja `Bitek` zwróciła wektor o innym rozmiarze niż $n \times n$.
- `WRONG_ANSWER 2` - któraś z wartości zwrócona przez `Bitek` nie jest z zakresu od 1 do 10^9 .
- `WRONG_ANSWER 3` - funkcja `Bajtazar` zwróciła wektor w którym występuje inna wartość niż 0, 1, 2, 3 lub 4.
- `WRONG_ANSWER 4` - jedna z dróg obranych przez Bajtazara jest nieoptymalna.

Kompilacja

Program **musi** zaczynać się od linijki `#include "wydlib.h"`

Program może używać zmiennych globalnych, ale ponieważ jest on uruchamiany oddzielnie dla Bitka i Bajtazara, uruchomienia te nie będą współdzielić zmiennych globalnych.

Sekcja studencka postanowiła udostępnić biblioteczkę testową służącą do zabawy. Może ona różnić się od biblioteczki używanej do oceniania rozwiązań.

Dla uczestników korzystających z systemu operacyjnego Linux (lub WSL itp.) - do czego bardzo gorąco zachęcamy - w zakładce Pliki zostało umieszczone archiwum `dlazaw.zip`, które należy pobrać i rozpakować.

Dla uczestników niemających dostępu do Linuxa udostępniamy środowisko w przeglądarce. Aby z niego skorzystać, należy wejść w link, założyć konto (jeśli takowego się jeszcze nie ma) i w prawym górnym rogu wybrać opcję `Use this template` -> `Open in codespaces`.

Swoje rozwiązanie należy umieścić w pliku `wyd.cpp`. By je skompilować, należy wykonać skrypt `./compile.sh`. Tworzy on plik `wyd.e`, który następnie należy uruchomić poleceniem `./wyd.e` i wpisać ręcznie test lub np. `./wyd.e < in/wyd0a.in`, żeby wczytać test z pliku.

Biblioteczka oczekuje najpierw liczby q a następnie q przypadków testowych. Na każdy przypadek testowy składają się:

- liczba n oznaczająca wymiary wydziału
- W K kolejnych wierszach współrzędne kolejnych sal
- w ostatnim wierszu para liczb a, b ($1 \leq a, b < n - 1$) oznaczająca początkową pozycję Bajtazara

Przykład

Wejście dla testu wyd0a:

```
1
5
1 1
1 2
1 3
2 1
2 2
2 3
3 1
2 2
```

Wyjaśnienie do przykładu: Testy przykładowe znajdują się w archiwum z przykładową biblioteką.

Sale z zajęciami znajdują się na środku wydziału. W tym przykładzie Bajtazar będzie w miejscu, z którego widać wszystkie sale z zajęciami (czego Bitek nie wie). Przykładowa strategia obrona przez Bitka polega na zaznaczeniu w każdej sali zajęciowej numeru zajęć. Strategia ta działa tylko jeśli Bajtazar przez przypadek będzie w dobrym miejscu.

Bitek zwraca:

```
{8,8,8,8,8},
{8,1,2,3,8},
{8,4,5,6,8},
{8,7,8,8,8},
{8,8,8,8,8} .
```

Wówczas Bajtazar widzi numery:

```
{1,2,3,
4,5,6,
7,8,8}
```

i wie, że jedną z poprawnych odpowiedzi jest {góra, góra, góra, lewo, zostań w miejscu, prawo, dół} czemu odpowiada wektor {3,3,3,1,4,0,2}.

Ocenianie

W tym zadaniu jest tylko jedna grupa testów. Czas na każdy test wynosi **2s**. Niech L oznacza największą wartość zwróconą przez Bitka.

Wynik jest obliczony następująco:

1. 0 punktów, jeśli $L > 10^9$.

2. 12 punktów, jeśli $7 \cdot 10^4 < L \leq 10^9$
3. 20 punktów, jeśli $10^4 < L \leq 7 \cdot 10^4$
4. 35 punktów, jeśli $2 \cdot 10^3 < L \leq 10^4$
5. $70 - 5.5 \cdot \log_{10} \left(\frac{L}{16} \right)$ punktów (zaokrąglone do liczby całkowitej), jeśli $16 < L \leq 2 \cdot 10^3$
6. 70 punktów, jeśli $L = 16$
7. 78 punktów, jeśli $L = 15$
8. 85 punktów, jeśli $L = 14$
9. 92 punktów, jeśli $L = 13$
10. 100 punktów, jeśli $L = 12$