

4. Mrówka Langtona

Zadanie

Opis problemu

Dana jest liczba całkowita n i kwadratowa plansza $n \times n$ pól. Proszę zaalokować dynamicznie pamięć dla planszy oraz zaimplementować działanie mrówki Langtona przy następujących założeniach:

1. Pola planszy przyjmują wartości całkowite od 1 do 8.
2. Krawędzie planszy są "sklejone" z przeciwległymi (periodyczne warunki brzegowe).
3. W chwili $t = 0$ mrówka stoi na polu o współrzędnych $[0][0]$ w kierunku pola $[0][1]$.
4. W chwili $t = 0$ pola planszy mają wartości losowe z podanego przedziału.
5. W zależności od stanu pola na którym stoi mrówka, wykonuje ona następujące czynności:

stan pola	zmiana stanu pola na	ruch mrówki
1	7	krok w przód
2	4	obrót w prawo
3	2	obrót w lewo
4	6	krok w prawy bok
5	3	krok w lewy bok
6	5	krok w tył
7	8	w tył zwrot
8	1	nic nie robi

Uzupełnij funkcje, których argumentem jest dwuwymiarowa tablica `board[] []` reprezentująca planszę, po której porusza się mrówka.

1. Funkcja `void init_board(int*** board, const int n)` alokuje pamięć dla kwadratowej tablicy `board` o wymiarach $[n][n]$ a następnie inicjalizuje ją liczbami losowymi z przedziału $[1, 8]$. Pola planszy wypełniane są wierszami.
2. Funkcja `void free_board(int **board, const int n)` zwalnia pamięć tablicy `board`.
3. Funkcja `void ant_move(int** board, const int n, Ant* ant)` wykonuje kolejny ruch mrówki zdefiniowanej w strukturze zgodnie z tabelą powyżej.

4. Funkcja `void ant_simulation(int** board, const int n, Ant* ant, const int steps)`, która inicjalizuje stan mrówki zgodnie z opisem a następnie wykonuje `steps` kroków (ruchów mrówki)
5. Funkcja `void print_board(int** board, const int n)` wypisuje stan planszy zgodnie z rosnącą wartością indeksów. Każde pole powinno być wypisane jako cyfra oznaczająca stan tego pola. Po każdym znaku pola powinna znajdować się spacja.
6. Funkcja `void print_ant(const Ant ant)` wypisuje stan mrówki (jej położenie i kierunek). Położenie należy wypisać jako parę liczb całkowitych będących indeksami pola, na którym stoi, a kierunek jest jedną z liter N, E, S, W (północ, wschód, południe, zachód). Przykład: jeżeli mrówka stoi na polu `[1][1]`, to jej kierunek określamy następująco:

mrówka patrzy w stronę	kierunek
<code>[0][1]</code>	N
<code>[1][2]</code>	E
<code>[2][1]</code>	S
<code>[1][0]</code>	W

Wejście

Trzy liczby całkowite: rozmiar planszy – liczba $10 \leq n \leq 100$, początkowa wartość ziarna generacji dla funkcji `srand()` i liczba kroków, $10 \leq s \leq 1000$.

Wyjście

Stan planszy po wykonaniu zadanej liczby kroków (funkcja `print_board()`) oraz końcowy stan mrówki (funkcja `print_ant()`).

Przykład:

Wejście:

10 5 20

Wyjście:

```

6 6 3 1 5 1 1 2 2 6
6 1 5 7 3 5 5 3 3 7
7 6 8 6 4 5 8 2 5 1
3 8 5 5 8 2 5 8 6 6
6 6 6 2 4 8 6 8 2 2
6 8 7 6 5 2 2 4 4 6
3 6 5 8 2 5 1 6 4 3
3 1 8 8 2 3 7 7 2 8
8 8 7 7 5 3 8 6 7 3
3 1 8 7 8 1 3 8 6 7
2 9 E

```