

### Задание. Вариант 3. Кратчайший путь

На шахматной доске размером  $N * N$  найти кратчайший путь ходами коня из поля А в поле В.

### Реализация на языке C.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_N 100

typedef struct {
    int row;
    int col;
} Point;

typedef struct {
    Point point;
    int dist;
} QueueNode;

// Возможные движения коня
int row_moves[] = {-2, -1, 1, 2, 2, 1, -1, -2};
int col_moves[] = {1, 2, 2, 1, -1, -2, -2, -1};

int isValid(int row, int col, int N) {
    return (row >= 0 && row < N && col >= 0 && col < N);
}

// Функция для поиска кратчайшего пути с использованием BFS
int bfs(Point start, Point end, int N) {
    int visited[MAX_N][MAX_N];
    memset(visited, 0, sizeof(visited));

    // Создаем очередь для BFS
    QueueNode queue[MAX_N * MAX_N];
    int front = 0, rear = 0;

    // Начальная позиция
    queue[rear++] = (QueueNode){start, 0};
    visited[start.row][start.col] = 1;

    while (front < rear) {
        QueueNode node = queue[front++];
        Point pt = node.point;

        // Если достигли конечной точки
        if (pt.row == end.row && pt.col == end.col) {
            return node.dist;
        }

        // Проверяем все возможные движения коня
        for (int i = 0; i < 8; i++) {
            int newRow = pt.row + row_moves[i];
            int newCol = pt.col + col_moves[i];
```

```

        if (isValid(newRow, newCol, N) && !visited[newRow][newCol]) {
            visited[newRow][newCol] = 1;
            queue[rear++] = (QueueNode){newRow, newCol, node.dist + 1};
        }
    }
}

// Если путь не найден
return -1;
}

int main() {
    int N;
    Point start, end;

    // Чтение размеров доски и начальной/конечной позиции
    printf("Введите размер доски N: ");
    scanf("%d", &N);

    printf("Введите начальную позицию коня (row col): ");
    scanf("%d %d", &start.row, &start.col);

    printf("Введите конечную позицию коня (row col): ");
    scanf("%d %d", &end.row, &end.col);

    int result = bfs(start, end, N);

    if (result != -1) {
        printf("Минимальное количество ходов: %d\n", result);
    } else {
        printf("Пути нет.\n");
    }

    return 0;
}

```

## Пример работы программы.

Взаимодействие с программой:

```

Введите размер доски N: 10
Введите начальную позицию коня (row col): 3 7
Введите конечную позицию коня (row col): 9 6
Минимальное количество ходов: 3

```