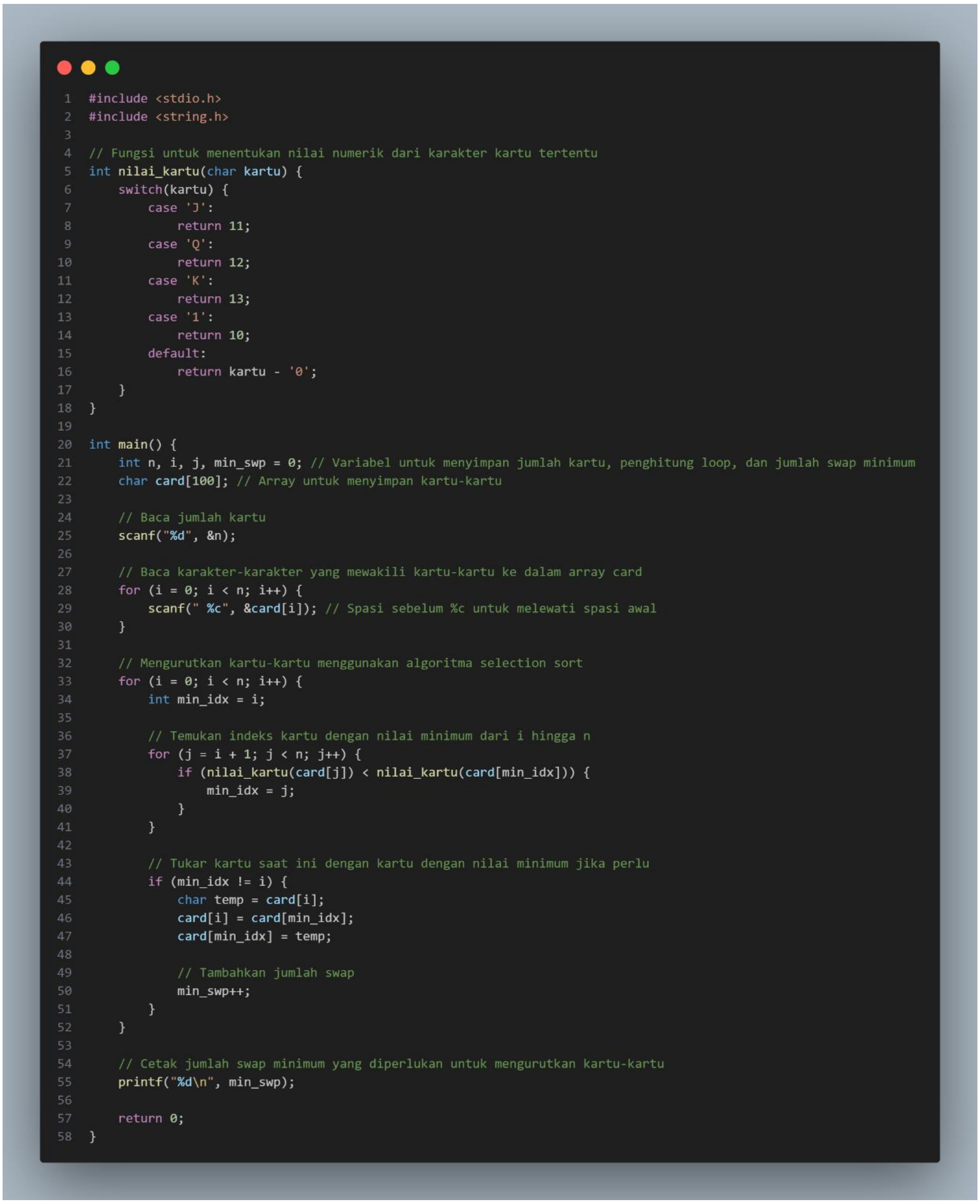
TUGAS OTH

NAMA : Al Fachri Sagianto

NIM : 1203230126

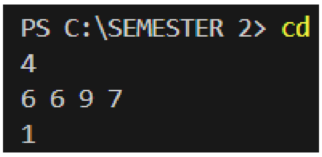
KELAS : IF-03-02

1. SOURCE CODE

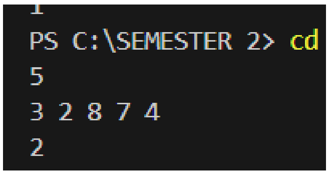


```c
#include <stdio.h>
#include <string.h>

// Fungsi untuk menentukan nilai numerik dari karakter kartu tertentu
int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

int main() {
    int n, i, j, min_swp = 0; // Variabel untuk menyimpan jumlah kartu, penghitung loop, dan jumlah swap minimum
    char card[100]; // Array untuk menyimpan kartu-kartu

    // Baca jumlah kartu
    scanf("%d", &n);

    // Baca karakter-karakter yang mewakili kartu-kartu ke dalam array card
    for (i = 0; i < n; i++) {
        scanf(" %c", &card[i]); // Spasi sebelum %c untuk melewati spasi awal
    }

    // Mengurutkan kartu-kartu menggunakan algoritma selection sort
    for (i = 0; i < n; i++) {
        int min_idx = i;

        // Temukan indeks kartu dengan nilai minimum dari i hingga n
        for (j = i + 1; j < n; j++) {
            if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {
                min_idx = j;
            }
        }

        // Tukar kartu saat ini dengan kartu dengan nilai minimum jika perlu
        if (min_idx != i) {
            char temp = card[i];
            card[i] = card[min_idx];
            card[min_idx] = temp;

            // Tambahkan jumlah swap
            min_swp++;
        }
    }

    // Cetak jumlah swap minimum yang diperlukan untuk mengurutkan kartu-kartu
    printf("%d\n", min_swp);

    return 0;
}
```

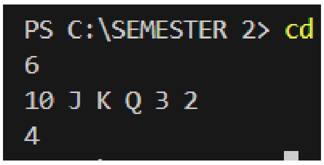
OUTPUT



```
PS C:\SEMESTER 2> cd
4
6 6 9 7
1
```

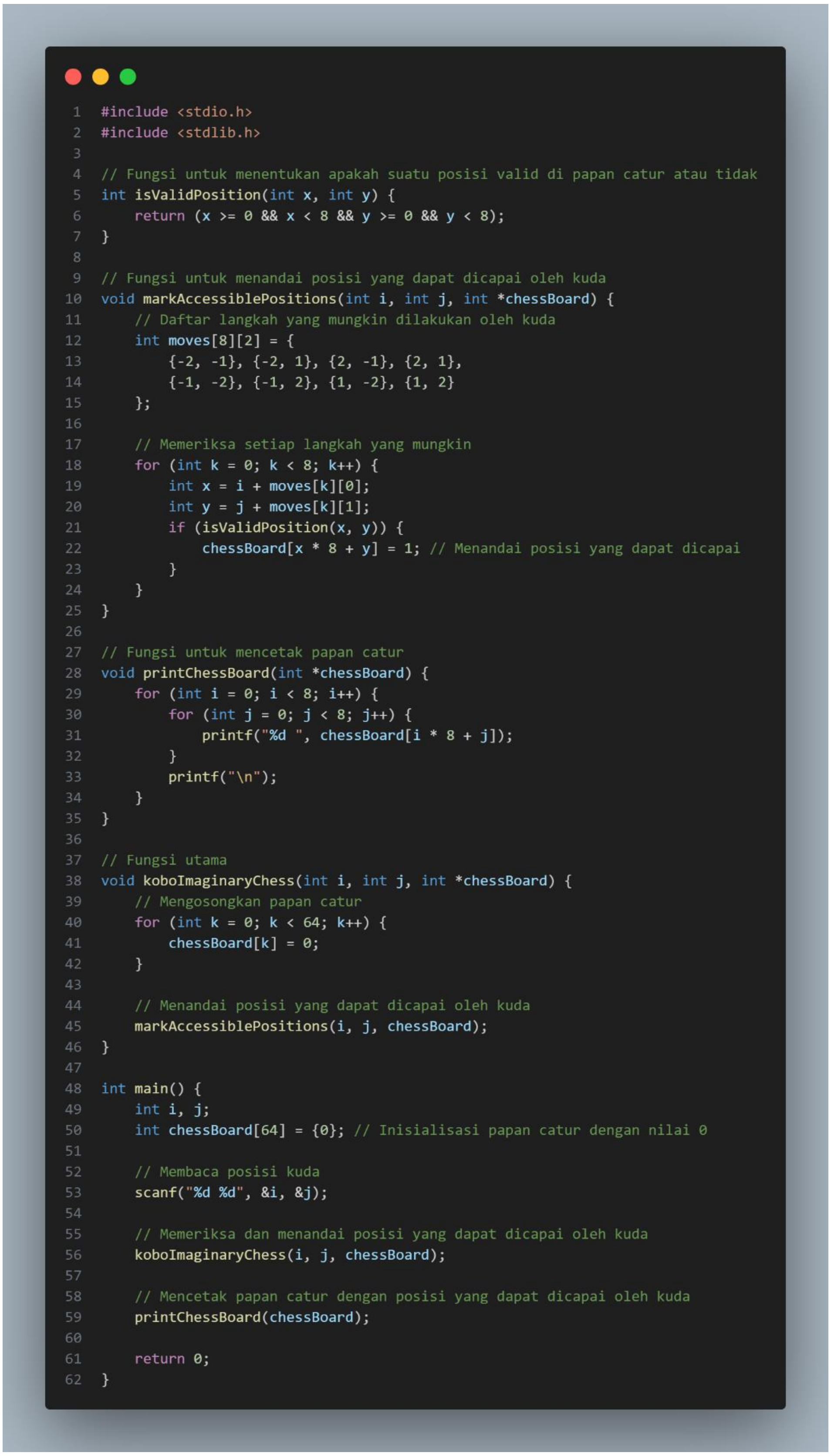


```
1
PS C:\SEMESTER 2> cd
5
3 2 8 7 4
2
```



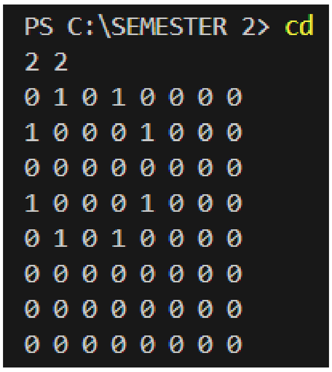
```
PS C:\SEMESTER 2> cd
6
10 J K Q 3 2
4
```

2. SOURCE CODE

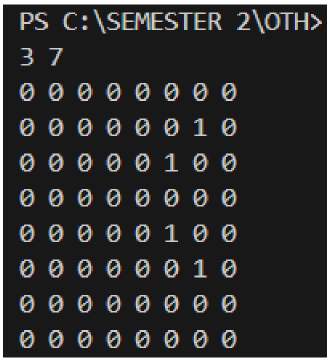


```c
#include <stdio.h>
#include <stdlib.h>

// Fungsi untuk menentukan apakah suatu posisi valid di papan catur atau tidak
int isValidPosition(int x, int y) {
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}

// Fungsi untuk menandai posisi yang dapat dicapai oleh kuda
void markAccessiblePositions(int i, int j, int *chessBoard) {
    // Daftar langkah yang mungkin dilakukan oleh kuda
    int moves[8][2] = {
        {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
        {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
    };

    // Memeriksa setiap langkah yang mungkin
    for (int k = 0; k < 8; k++) {
        int x = i + moves[k][0];
        int y = j + moves[k][1];
        if (isValidPosition(x, y)) {
            chessBoard[x * 8 + y] = 1; // Menandai posisi yang dapat dicapai
        }
    }
}

// Fungsi untuk mencetak papan catur
void printChessBoard(int *chessBoard) {
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            printf("%d ", chessBoard[i * 8 + j]);
        }
        printf("\n");
    }
}

// Fungsi utama
void koboImaginaryChess(int i, int j, int *chessBoard) {
    // Mengosongkan papan catur
    for (int k = 0; k < 64; k++) {
        chessBoard[k] = 0;
    }

    // Menandai posisi yang dapat dicapai oleh kuda
    markAccessiblePositions(i, j, chessBoard);
}

int main() {
    int i, j;
    int chessBoard[64] = {0}; // Inisialisasi papan catur dengan nilai 0

    // Membaca posisi kuda
    scanf("%d %d", &i, &j);

    // Memeriksa dan menandai posisi yang dapat dicapai oleh kuda
    koboImaginaryChess(i, j, chessBoard);

    // Mencetak papan catur dengan posisi yang dapat dicapai oleh kuda
    printChessBoard(chessBoard);

    return 0;
}
```

OUTPUT

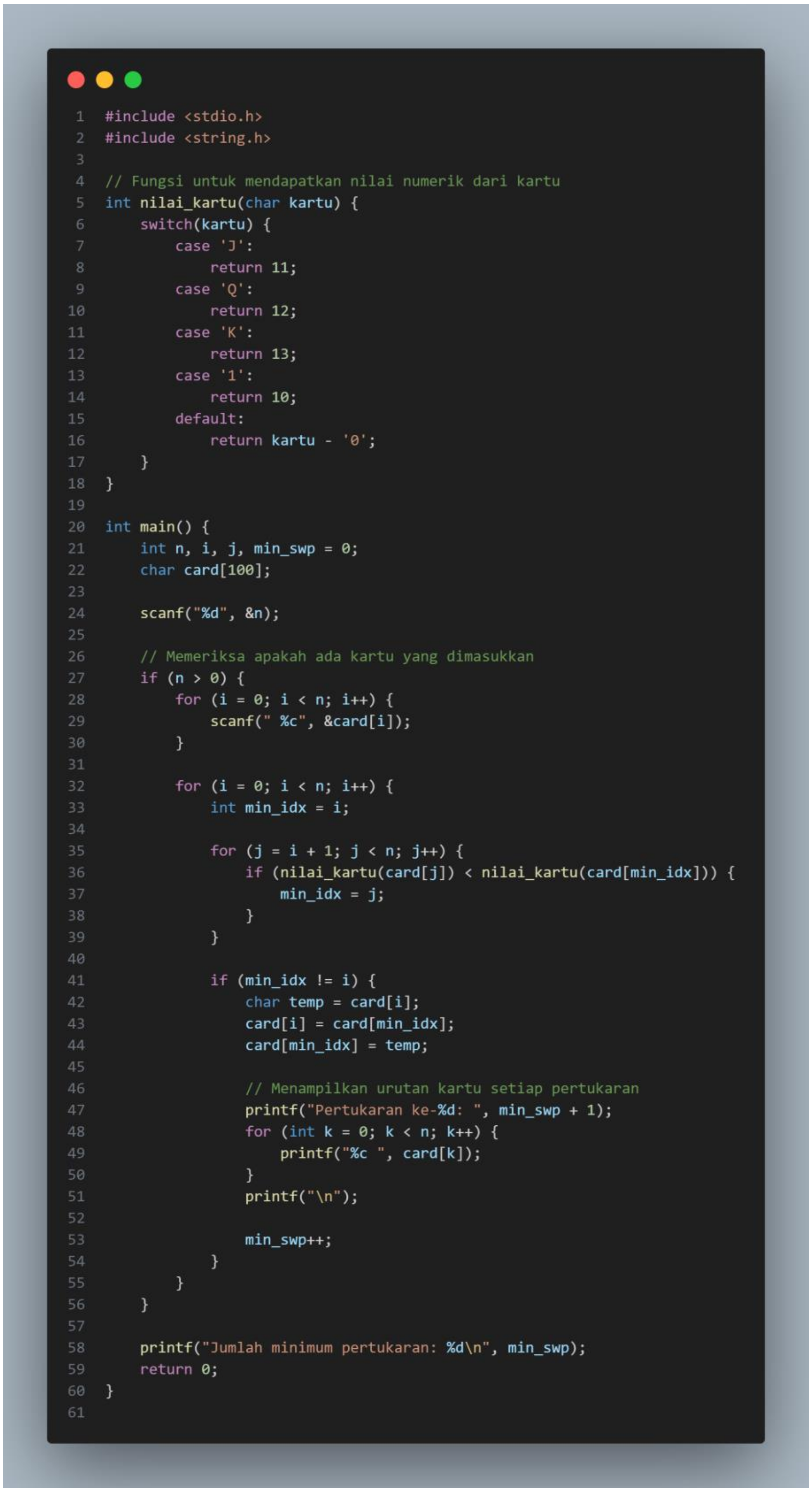


```
PS C:\SEMESTER 2> cd
2 2
0 1 0 1 0 0 0 0
1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0
0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```



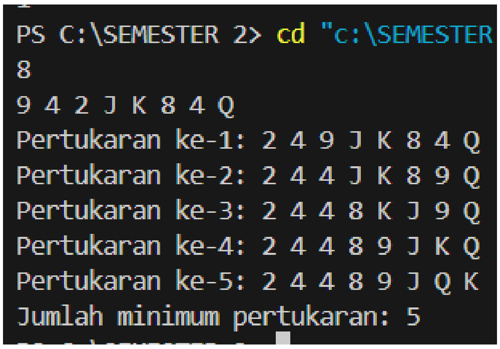
```
PS C:\SEMESTER 2\OTH>
3 7
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

CODE BONUS SOAL



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

// Fungsi untuk mendapatkan nilai numerik dari kartu
int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

int main() {
    int n, i, j, min_swp = 0;
    char card[100];

    scanf("%d", &n);

    // Memeriksa apakah ada kartu yang dimasukkan
    if (n > 0) {
        for (i = 0; i < n; i++) {
            scanf(" %c", &card[i]);
        }

        for (i = 0; i < n; i++) {
            int min_idx = i;

            for (j = i + 1; j < n; j++) {
                if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {
                    min_idx = j;
                }
            }

            if (min_idx != i) {
                char temp = card[i];
                card[i] = card[min_idx];
                card[min_idx] = temp;

                // Menampilkan urutan kartu setiap pertukaran
                printf("Pertukaran ke-%d: ", min_swp + 1);
                for (int k = 0; k < n; k++) {
                    printf("%c ", card[k]);
                }
                printf("\n");

                min_swp++;
            }
        }
    }

    printf("Jumlah minimum pertukaran: %d\n", min_swp);
    return 0;
}

```

OUTPUT



```
PS C:\SEMESTER 2> cd "c:\SEMESTER
8
9 4 2 J K 8 4 Q
Pertukaran ke-1: 2 4 9 J K 8 4 Q
Pertukaran ke-2: 2 4 4 J K 8 9 Q
Pertukaran ke-3: 2 4 4 8 K J 9 Q
Pertukaran ke-4: 2 4 4 8 9 J K Q
Pertukaran ke-5: 2 4 4 8 9 J Q K
Jumlah minimum pertukaran: 5
```