

## Praktikum ASD

Nama: Gibraltar Aldebran Abdallah Dj

Nim: 1203230033

Kelas: IF-03-02

### 1.Source Code

```
#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;

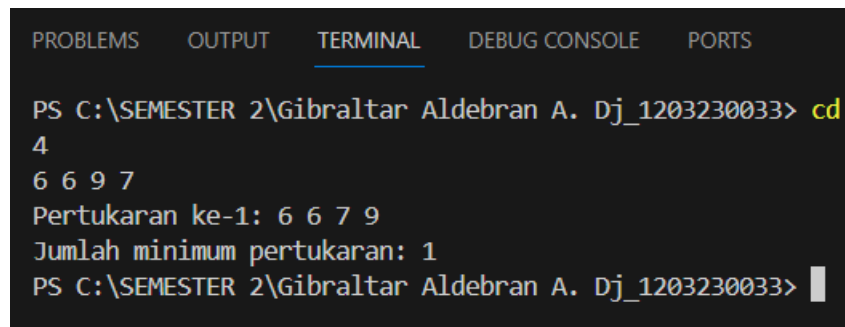
            // Menampilkan urutan kartu setiap pertukaran
            printf("Pertukaran ke-%d: ", min_swp + 1);
            for (int k = 0; k < n; k++) {
                printf("%c ", card[k]);
            }
            printf("\n");

            min_swp++;
        }
    }

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

```

## Output



```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

PS C:\SEMESTER 2\Gibraltar Aldebran A. Dj_1203230033> cd
4
6 6 9 7
Pertukaran ke-1: 6 6 7 9
Jumlah minimum pertukaran: 1
PS C:\SEMESTER 2\Gibraltar Aldebran A. Dj_1203230033>

```

## Penjelasan

### 1. Fungsi `nilai_kartu(char kartu)`:

- Fungsi ini mengambil karakter kartu sebagai input dan mengembalikan nilai numerik dari kartu tersebut.
- Jika karakter kartu adalah 'J', 'Q', atau 'K', maka fungsi akan mengembalikan nilai 11, 12, atau 13.
- Jika karakter kartu adalah angka (1-9), maka fungsi akan mengembalikan nilai angka tersebut.

### 2. Variabel `n`, `i`, `j`, dan `min_swp`:

- **n** adalah jumlah kartu yang akan dimasukkan.
  - **i, j,** dan **min\_swp** adalah variabel indeks yang digunakan untuk mengakses elemen dalam array **card**.
3. Perulangan **for** dan **if** untuk mencari indeks minimum:
- Perulangan **for** di dalam blok kode ini mengulangi sebanyak **n** kali untuk mencari indeks kartu dengan nilai minimum.
  - Dalam setiap iterasi, variabel **min\_idx** akan menyimpan indeks kartu dengan nilai minimum.
  - Jika ditemukan kartu dengan nilai yang lebih kecil daripada kartu pada indeks **min\_idx**, maka **min\_idx** akan diperbarui dengan indeks kartu yang baru ditemukan.
4. Perulangan **for** untuk menukar kartu:
- Jika ditemukan indeks kartu dengan nilai yang lebih kecil daripada kartu pada indeks **i**, maka k

## 2.Source Code

```
#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

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

PS C:\SEMESTER 2\Gibraltar Aldebran A. Dj_1203230033> 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\Gibraltar Aldebran A. Dj_1203230033> |
```

## Penjelasan

Dalam fungsi **isValidPosition**, kita memeriksa apakah posisi yang diberikan valid atau tidak. Posisi valid adalah posisi yang berada di antara indeks 0 hingga 7 (termasuk) di baris dan kolom.

Dalam fungsi **markAccessiblePositions**, kita menggunakan array **moves** yang berisi 8 langkah yang mungkin dilakukan oleh kuda. Kemudian, kita memeriksa setiap langkah yang mungkin dan menandai posisi yang dapat dicapai oleh kuda.

Dalam fungsi **kobolmaginaryChess**, kita mengosongkan papan catur dengan nilai 0 dan menandai posisi yang dapat dicapai oleh kuda.

Dalam fungsi **printChessBoard**, kita mencetak papan catur dengan posisi yang dapat dicapai oleh kuda.

Dalam fungsi **main**, kita membaca posisi kuda dari inputan pengguna dan menjalankan fungsi **kobolmaginaryChess** untuk mencari dan menandai posisi yang dapat dicapai oleh kuda. Kemudian, kita mencetak papan catur dengan posisi yang dapat dicapai oleh kuda.