

Nama : Khusnia Fitri

NIM : 1203230030

Komponen Penilaian	Ya	Tidak
Soal 1 sesuai dengan output yang		
Diinginkan Soal 2 sesuai dengan output		
Yang diinginkan Bonus soal 1 dikerjakan		

## 1. Kode

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

int convertToNumericValue(char card[]) {
    if (strcmp(card, "10") == 0) {
        return 10;
    } else if (strlen(card) == 1) {
        if (card[0] >= '2' && card[0] <= '9') {
            return card[0] - '0';
        } else if (card[0] == 'J' || card[0] == 'j') {
            return 11;
        } else if (card[0] == 'Q' || card[0] == 'q') {
            return 12;
        } else if (card[0] == 'K' || card[0] == 'k') {
            return 13;
        }
    }

    printf("Nilai kartu tidak valid: %s\n", card);
    exit(EXIT_FAILURE);
}

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int countMinimumSteps(int cards[], int n) {
    int steps = 0;
    int i, j, min_index;

    for (i = 0; i < n - 1; i++) {
```

```

        min_index = i;
        for (j = i + 1; j < n; j++) {
            if (cards[j] < cards[min_index]) {
                min_index = j;
            }
        }
        if (min_index != i) {
            swap(&cards[i], &cards[min_index]);
            steps++;
        }
    }

    return steps;
}

int main() {
    int n;
    printf("Jumlah kartu: ");
    scanf("%d", &n);

    char inputCards[n][3];
    int numericCards[n];

    printf("Masukkan nilai kartu: ");
    for (int i = 0; i < n; i++) {
        scanf("%s", inputCards[i]);
        numericCards[i] = convertToNumericValue(inputCards[i]);
    }

    int steps = countMinimumSteps(numericCards, n);

    printf("%d\n", n);
    for (int i = 0; i < n; i++) {
        printf("%s ", inputCards[i]);
    }
    printf("\n");

    printf("%d\n", steps);

    return 0;
}

```

**Penjelasan :**

1. **convertToNumericValue:** Fungsi ini mengonversi nilai kartu dalam bentuk karakter menjadi nilai numerik. Misalnya, dalam kartu terdapat “J” “Q” “K”, dimana: J=10, Q=12, K=13.
2. **swap(int a, int b):** Fungsi ini digunakan untuk menukar dua nilai integer. Ini akan digunakan menggunakan algoritma pengurutan Selection Sort untuk menukar posisi kartu dari yang terkecil.
3. **countMinimumSteps:** Fungsi yang menghitung jumlah langkah/step pertukaran yang diperlukan untuk mengurutkan menggunakan selection sort.
4. **main():** Berisi untuk memasukkan jumlah kartu dan nilai kartu yang akan di simpan dalam array **inputCards**(Array dua dimensi yang digunakan untuk menyimpan nilai kartu yang dimasukkan oleh pengguna dalam bentuk karakter) dan **numericCards**(Setiap nilai kartu yang dimasukkan diubah menjadi nilai numerik menggunakan fungsi **convertToNumericValue** dan disimpan dalam array ini). Selanjutnya, program menghitung jumlah langkah minimal untuk mengurutkan kartu menggunakan Selection Sort dan mencetak jumlah kartu, nilai kartu sebelum diurutkan, dan jumlah langkah yang diperlukan.

#### Output :

```
Jumlah kartu: 4
Masukkan nilai kartu: 6 6 9 7
4
6 6 9 7
1
```

```
Jumlah kartu: 5
Masukkan nilai kartu: 3 2 8 7 4
5
3 2 8 7 4
2
```

```
Jumlah kartu: 6
Masukkan nilai kartu: 10 J K Q 3 2
6
10 J K Q 3 2
4
```

#### Bonus :

#### 2. kode :

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            *(chessBoard + row * size + col) = 0;
        }
    }

    int moves[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1}, {2, 1}, {1, 2},
{-1, 2}, {-2, 1}};
    for (int k = 0; k < 8; k++) {
        int newRow = i + moves[k][0];
```

```

        int newCol = j + moves[k][1];
        if (newRow >= 0 && newRow < size && newCol >= 0 && newCol < size) {
            *(chessBoard + newRow * size + newCol) = 1;
        }
    }

    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            printf("%d ", *(chessBoard + row * size + col));
        }
        printf("\n");
    }
}

int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int size = 8;
    int chessBoard[size][size];
    koboImaginaryChess(i, j, size, (int *)chessBoard);
    return 0;
}

```

output :

2 2	3 7
0 1 0 1 0 0 0 0	0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0	0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0
1 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0
0 1 0 1 0 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 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 0 0 0 0 0

### Penjelasan :

fungsi **koboImaginaryChess** yang menerima input koordinat posisi kuda (i dan j) serta ukuran papan catur (yang dalam kasus ini adalah 8). Fungsi tersebut akan mengisi sebuah array 2D (papan catur) dengan nilai awal 0 di setiap kotaknya. Kemudian, fungsi tersebut akan menandai posisi-posisi yang dapat dicapai oleh kuda dalam satu langkah dengan memberi nilai 1 pada kotak-kotak tersebut. Program utama akan membaca input koordinat posisi kuda, kemudian memanggil fungsi **koboImaginaryChess** untuk menampilkan hasilnya.