

1

```
#include <stdio.h>

// Fungsi untuk menukar dua nilai
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Fungsi untuk mengonversi kartu menjadi nilai numerik
int valuekartu(char kartu[], int i) {
    if (kartu[i] == 'J' || kartu[i] == 'j') {
        return 11;
    } else if (kartu[i] == 'Q' || kartu[i] == 'q') {
        return 12;
    } else if (kartu[i] == 'K' || kartu[i] == 'k') {
        return 13;
    } else {
        return kartu[i] - '0';
    }
}

// Fungsi partisi untuk algoritma quicksort
int partition(int arr[], int low, int high, int *steps) {
    int pivot = arr[high]; // Mengambil pivot sebagai elemen terakhir
    int i = (low - 1);

    // Iterasi melalui elemen-elemen array
    for (int j = low; j <= high - 1; j++) {
        // Jika elemen saat ini lebih kecil dari pivot, tukar dengan elemen yang lebih
        // besar
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]); // Menukar elemen
            (*steps)++; // Menghitung langkah pertukaran
            printf("Pertukaran %d: ", *steps);
            for (int k = 0; k <= high; k++) {
                printf("%d ", arr[k]);
            }
            printf("\n");
        }
    }
    swap(&arr[i + 1], &arr[high]); // Tukar elemen pivot ke posisi yang tepat
    return (i + 1); // Kembalikan indeks pivot
}
```

```

// Fungsi rekursif untuk quicksort
void quickSort(int arr[], int low, int high, int *steps) {
    if (low < high) {
        // Bagi array dan dapatkan indeks partisi
        int pi = partition(arr, low, high, steps);
        // Panggil quickSort untuk bagian kiri dan kanan dari partisi
        quickSort(arr, low, pi - 1, steps);
        quickSort(arr, pi + 1, high, steps);
    }
}

int main() {
    int jumlah;
    scanf("%i", &jumlah);

    char kartu[50][2];
    int value[50];

    // Meminta input kartu
    for (int i = 0; i < jumlah; i++) {
        scanf("%s", kartu[i]);
        // Konversi kartu menjadi nilai numerik
        if (kartu[i][0] == '1' && kartu[i][1] == '0') {
            value[i] = 10;
        } else {
            value[i] = valuekartu(kartu[i], i);
        }
    }

    int langkah = 0;
    // Panggil quickSort untuk mengurutkan nilai numerik
    quickSort(value, 0, jumlah - 1, &langkah);

    printf("%i", langkah);
    return 0;
}

```

```

1
● mac@macs-MacBook-Air ASD % cd "/Users/m
D/Praktek 4/TUGAS OTH/"soal1
4
6 6 9 7
Pertukaran 1: 6 6 9 7
Pertukaran 2: 6 6 9 7
2%
● mac@macs-MacBook-Air TUGAS OTH % cd "/U
T 2/ASD/Praktek 4/TUGAS OTH/"soal1
5
3 2 8 7 4
Pertukaran 1: 3 2 8 7 4
Pertukaran 2: 3 2 8 7 4
Pertukaran 3: 2 3 4 7 8
3%
● mac@macs-MacBook-Air TUGAS OTH % cd "/U
T 2/ASD/Praktek 4/TUGAS OTH/"soal1
6
10 j k q 3 2
Pertukaran 1: 2 3 13 12 11 10
Pertukaran 2: 2 3 10 12 11 13
Pertukaran 3: 2 3 10 12 11 13
3%
● mac@macs-MacBook-Air TUGAS OTH %

```

2

```

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // fungsi yang menentukan benar atau tidak nya posisi kuda
    if (i < 0 || j < 0 || i >= size || j >= size) {
        printf("Koordinat tidak valid\n");
        return;
    }

    //ini merupakan semua kemungkinan perpindahan
    int pindah[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1}, {-2, 1}, {-1, 2}, {1,
2}, {2, 1}}; //algoritma penempatan kuda

    //ini merupakan loop yang digunakan untuk menyatakan semua kemungkinan diatas lalu
ditandai dengan 1
    for (int k = 0; k < 8; k++) {
        int ni = i + pindah[k][0];
        int nj = j + pindah[k][1];
        if (ni >= 0 && nj >= 0 && ni < size && nj < size) {

```

```

        chessBoard[ni * size + nj] = 1;
    }
}

int main() {
    int catur[8][8] = {0};

    int i, j;
    scanf("%d %d", &i, &j);

    koboImaginaryChess(i, j, 8, (int *)catur);

    //menampilkan papan catur yang telah di perbarui
    for (int a = 0; a < 8; a++) {
        for (int b = 0; b < 8; b++) {
            printf("%d ", catur[a][b]);
        }
        printf("\n");
    }

    return 0;
}

```

```

2
● mac@macs-MacBook-Air ASD % cd "D/Praktek 4/TUGAS 0TH/"soal2
3 3
0 0 0 0 0 0 0 0
0 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
● mac@macs-MacBook-Air TUGAS 0TH % cd "T 2/ASD/Praktek 4/TUGAS 0TH/"soal2
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
○ mac@macs-MacBook-Air TUGAS 0TH %

```

Komponen penilaian	Ya	Tidak
Soal 1 sesuai dengan output yang diinginkan		
Soal 2 sesuai dengan output yang diinginkan		
Bonus soal 1 dikerjakan		