

NAMA : FITRIA NUR ROFIKA

NIM : 1203230097

1.

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

int card_value(char card) {
    if (card >= '2' && card <= '9') {
        return card - '0'
    } else if (card == 'J') {
        return 11;
    } else if (card == 'Q') {
        return 12;
    } else if (card == 'K') {
        return 13;
    } else {
        return 10;
    }
}

void swap(char *x, char *y) {
    char temp = *x;
    *x = *y;
    *y = temp;
}

int bubble_sort(char arr[], int n) {
    int i, j, steps = 0;
    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (card_value(arr[j]) > card_value(arr[j+1])) {
                swap(&arr[j], &arr[j+1]);
                steps++;
            }
        }
    }
    return steps;
}

int main() {
    int n, steps;
    char cards[100];

    scanf("%d", &n);
```

```

    getchar();    fgets(cards, sizeof(cards), stdin);

    steps = bubble_sort(cards, n);

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

    return 0;
}

```

OUTPUT

```

PS D:\TUGAS TELKOM\algoritma\praktikum\praktikum> cd "d:\TUGAS TELKOM\algoritma\praktikum\praktikum" ; if ($?) { gcc tempCodeRunnerFile.c -o te
le } ; if ($?) { .\tempCodeRunnerFile }
4
6 6 9 7
1
PS D:\TUGAS TELKOM\algoritma\praktikum\praktikum>

```

PENJELASAN

Program ini untuk menyelesaikan masalah Refan yang ingin mengurutkan kartu yang ia miliki dari terkecil hingga terbesar. Program menggunakan Bubble Sort untuk mengurutkan kartu.

1. Program dimulai dengan.
 - swap: Fungsi untuk menukar nilai dua kartu.
 - bubble_sort: Fungsi untuk mengurutkan kartu dan menghitung jumlah pertukaran.
2. Bubble Sort digunakan untuk mengurutkan kartu. Setiap kali ada pertukaran, urutan kartu baru akan ditampilkan. Jumlah pertukaran yang terjadi juga dihitung.
3. Di dalam main function:
 - User diminta untuk memasukkan jumlah kartu (n) dan nilai-nilai kartu yang tidak terurut.
 - Kartu-kartu tersebut disimpan dalam array cards.
 - Array cards kemudian diurutkan menggunakan fungsi bubble_sort.
 - Jumlah minimal langkah pertukaran yang diperlukan untuk mengurutkan kartu ditampilkan.
4. Program mengambil input jumlah kartu dan nilai-nilai kartu dari pengguna. Kemudian, algoritma Bubble Sort dijalankan untuk mengurutkan kartu dan menampilkan urutan setiap kali terjadi pertukaran.

```

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

int isValidPosition(int x, int y) {
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}

void markAccessiblePositions(int i, int j, int *chessBoard) {
    int moves[8][2] = {

```

```

        {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
        {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
    };

    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;
        }
    }
}

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");
    }
}

void koboImaginaryChess(int i, int j, int *chessBoard) {
    for (int k = 0; k < 64; k++) {
        chessBoard[k] = 0;
    }
    markAccessiblePositions(i, j, chessBoard);
}

int main() {
    int i, j;
    int chessBoard[64] = {0};
    scanf("%d %d", &i, &j);

    koboImaginaryChess(i, j, chessBoard);
    printChessBoard(chessBoard);

    return 0;
}

```

OUTPUT

Penjelasan mengenai program:2

1. Fungsi koboImaginaryChess menerima input berupa posisi awal kuda (i, j), ukuran papan catur (size), dan array chessBoard yang merepresentasikan papan catur. Fungsi ini menggunakan daftar langkah yang mungkin dilakukan oleh kuda untuk menandai posisi-posisi yang dapat dicapai oleh kuda dalam sekali jalan.

2. Di dalam main function, program membaca input posisi awal kuda dari pengguna. Kemudian, program memanggil fungsi `kobolmaginaryChess` untuk menandai posisi yang dapat dicapai oleh kuda.

3. Setelah itu, program mencetak papan catur yang telah dimodifikasi ke layar. Setiap posisi yang dapat dicapai oleh kuda akan ditandai dengan nilai 1, sedangkan posisi lainnya akan tetap bernilai 0.

KOMPONEN PENILAIAN	YA	TIDAK
Soal 1 sesuai dengan output yang diinginkan	✓	
Soal 2 sesuai dengan output yang diinginkan	✓	
BONUS		✓