Tugas praktikum Algoritma dan struktur data IF-03-03

Nama : Rahmadi Rafiansyah

Nim : 1203230075

Kelas : IF 03-03

1. Screenshot full card game

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

// Fungsi untuk mengimplementasikan nilai kartu yaitu J, Q, K //
int compare(const void *a, const void *b) {
    char cardA = *(char *)a;
    char cardB = *(char *)b;
    char order[] = "123456789JQK";

    int indexA = strchr(order, cardA) - order;
    int indexB = strchr(order, cardB) - order;

    return indexA - indexB;
}

// Selection sort untuk mengurutkan kartu //
int sortCards(int n, char cards[]) {
    int steps = 0;
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (compare(&cards[j], &cards[minIndex]) < 0) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            char temp = cards[i];
            cards[i] = cards[minIndex];
            cards[minIndex] = temp;
            steps++;
            // Menampilkan setiap pertukaran yang terjadi //
            printf("Pertukaran %d: ", steps);
            for (int k = 0; k < n; k++) {
                printf("%c ", cards[k]);
            }
            printf("\n");
        }
    }
    return steps;
}

int main() {
    int n;
    // Inputan jumlah kartu //
    printf("");
    scanf("%d", &n);

    char cards[n];
    // Memasukkan kartu //
    printf("");
    for (int i = 0; i < n; i++) {
        scanf(" %c", &cards[i]);
    }

    int steps = sortCards(n, cards);

    // Mencetak kartu yang telah diurutkan //
    printf("Jumlah langkah untuk mengurutkan kartu : %d\n", steps);
    printf("Urutan kartu setelah diurutkan : ");
    for (int i = 0; i < n; i++) {
        printf("%c ", cards[i]);
    }
    printf("\n");

    return 0;
}
```

Fungsi compare :

- Fungsi ini digunakan sebagai fungsi pembanding untuk digunakan dalam proses pengurutan.
- Mengambil dua pointer ke karakter (karena kartu adalah karakter) sebagai argumen.
- Membandingkan kartu-kartu berdasarkan urutan yang telah ditentukan: "123456789JQK".
- Mengembalikan nilai positif jika kartu pertama harus berada sebelum kartu kedua, nilai negatif jika sebaliknya, dan nol jika kartu-kartu tersebut sama.

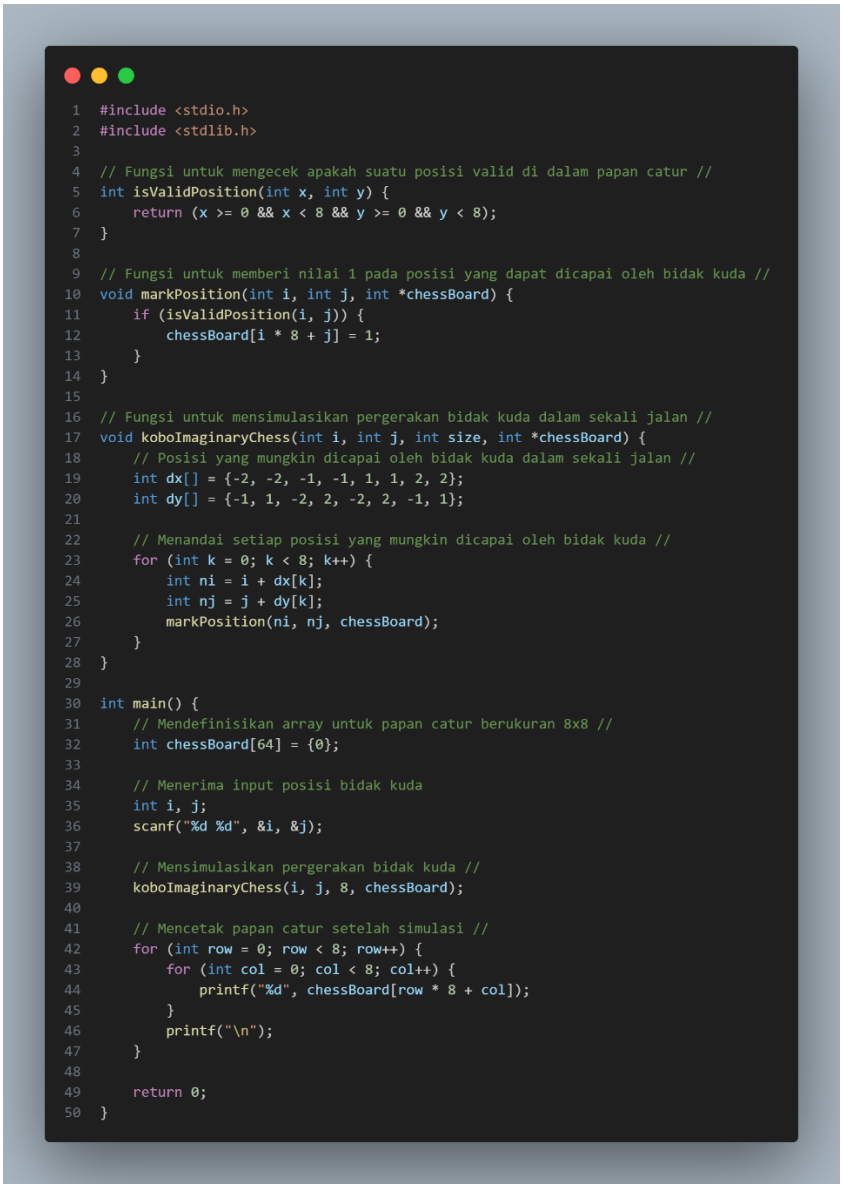
Fungsi sortCards :

- Menerima jumlah kartu n dan array cards yang berisi karakter-karakter kartu.
- Melakukan pengurutan kartu menggunakan algoritma selection sort.
- Selama proses pengurutan, setiap kali ada pertukaran kartu, jumlah langkah (steps) akan bertambah satu.
- Setelah pengurutan selesai, fungsi ini mengembalikan jumlah langkah yang diperlukan untuk mengurutkan kartu.

Fungsi main :

- Mengambil input jumlah kartu n dari pengguna.
- Mengambil input nilai kartu satu per satu dan menyimpannya dalam array cards.
- Memanggil fungsi sortCards untuk mengurutkan kartu.
- Setelah pengurutan selesai, mencetak jumlah langkah yang diperlukan dan urutan kartu yang telah diurutkan.

2. Screenshot full chess game



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

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

// Fungsi untuk memberi nilai 1 pada posisi yang dapat dicapai oleh bidak kuda //
void markPosition(int i, int j, int *chessBoard) {
    if (isValidPosition(i, j)) {
        chessBoard[i * 8 + j] = 1;
    }
}

// Fungsi untuk mensimulasikan pergerakan bidak kuda dalam sekali jalan //
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Posisi yang mungkin dicapai oleh bidak kuda dalam sekali jalan //
    int dx[] = {-2, -2, -1, -1, 1, 1, 2, 2};
    int dy[] = {-1, 1, -2, 2, -2, 2, -1, 1};

    // Menandai setiap posisi yang mungkin dicapai oleh bidak kuda //
    for (int k = 0; k < 8; k++) {
        int ni = i + dx[k];
        int nj = j + dy[k];
        markPosition(ni, nj, chessBoard);
    }
}

int main() {
    // Mendefinisikan array untuk papan catur berukuran 8x8 //
    int chessBoard[64] = {0};

    // Menerima input posisi bidak kuda
    int i, j;
    scanf("%d %d", &i, &j);

    // Mensimulasikan pergerakan bidak kuda //
    koboImaginaryChess(i, j, 8, chessBoard);

    // Mencetak papan catur setelah simulasi //
    for (int row = 0; row < 8; row++) {
        for (int col = 0; col < 8; col++) {
            printf("%d", chessBoard[row * 8 + col]);
        }
        printf("\n");
    }

    return 0;
}
```

Fungsi isValidPosition :

- Fungsi ini digunakan untuk memeriksa apakah suatu posisi (x, y) berada di dalam papan catur yang valid.
- Mengembalikan nilai 1 jika posisi valid, dan 0 jika tidak.

Fungsi markPosition :

- Fungsi ini digunakan untuk memberi nilai 1 pada posisi yang dapat dicapai oleh bidak kuda.
- Menerima input koordinat (i, j) dan array chessBoard yang merepresentasikan papan catur.
- Jika posisi (i, j) valid, maka nilai pada indeks yang sesuai di array chessBoard diubah menjadi 1.

Fungsi koboImaginaryChess :

- Fungsi ini mensimulasikan pergerakan bidak kuda dalam sekali jalan.
- Menerima input posisi awal bidak kuda (i, j), ukuran papan catur (size), dan array chessBoard.
- Menggunakan array dx dan dy untuk menyimpan perubahan posisi yang mungkin dari posisi awal bidak kuda.
- Memanggil fungsi markPosition untuk menandai setiap posisi yang dapat dicapai oleh bidak kuda.

Fungsi main :

- Mendefinisikan array chessBoard untuk merepresentasikan papan catur berukuran 8x8 dengan nilai awal 0 di setiap indeks.
- Menerima input posisi awal bidak kuda (i, j).
- Memanggil fungsi koboImaginaryChess untuk mensimulasikan pergerakan bidak kuda.
- Mencetak papan catur setelah simulasi, di mana nilai 1 menunjukkan posisi yang dapat dicapai oleh bidak kuda, dan nilai 0 menunjukkan posisi yang tidak dapat dicapai.