Benediktus Daniel Afriant

1203230005

Informatika 03-03

1. Kartu

Source Code :

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

// Fungsi untuk mengonversi kartu menjadi angka (untuk mempermudah
perbandingan)
int convertCardValue(char card) {
    if (card >= '2' && card <= '9')
        return card - '0';
    else if (card == '1' || card == 'J')
        return 10;
    else if (card == 'Q')
        return 11;
    else if (card == 'K')
        return 12;
    return -1; // Kartu invalid
}

// Fungsi untuk mengurutkan kartu menggunakan algoritma Bubble Sort
int bubbleSort(int *cards, int n) {
    int steps = 0;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (cards[j] > cards[j + 1]) {
                // Swap cards
                int temp = cards[j];
                cards[j] = cards[j + 1];
                cards[j + 1] = temp;

                // Menghitung Pengurutan
                steps++;

                // Menampilkan urutan setiap perulangan
                printf("Pertukaran %d: ", steps);
                for (int k = 0; k < n; k++) {
                    printf("%d ", cards[k]);
                }
                printf("\n");
            }
        }
```

```
    }
    return steps;
}

int main() {
    int N;
    scanf("%d", &N);

    // Membaca kartu-kartu yang dimiliki Refan
    int *cards = (int *)malloc(N * sizeof(int));
    char input[3];
    for (int i = 0; i < N; i++) {
        scanf("%s", input);
        cards[i] = convertCardValue(input[0]);
    }

    // Mengurutkan kartu menggunakan Bubble Sort dan menghitung jumlah langkah
pertukaran
    int minSteps = bubbleSort(cards, N);

    printf("Jumlah langkah pertukaran: %d\n", minSteps);

    free(cards); // Membebaskan memori yang dialokasikan untuk array kartu
    return 0;
}
```

Output :

```
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test 03.exe'
5
9 3 6 8 5
Langkah 1: 3 9 6 8 5
Langkah 2: 3 6 9 8 5
Langkah 3: 3 6 8 9 5
Langkah 4: 3 6 8 5 9
Langkah 5: 3 6 5 8 9
Langkah 6: 3 5 6 8 9
Jumlah langkah pertukaran minimal: 6
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test 03.exe'cd 'c:\Users\Daniel\Downloads\PS1\output'
```

Penjelasan :

1. #include <stdio.h>: Baris ini memasukkan file header stdio.h yang diperlukan untuk fungsi input-output standar dalam bahasa C.
2. #include <stdlib.h>: Baris ini memasukkan file header stdlib.h yang diperlukan untuk alokasi memori dinamis menggunakan malloc() dan free().
3. #include <string.h>: Baris ini memasukkan file header string.h yang diperlukan untuk beberapa fungsi pengolahan string.

4. int convertCardValue(char card) { ... }: Fungsi ini mengonversi nilai kartu menjadi angka untuk mempermudah perbandingan. Misalnya, kartu '2' menjadi angka 2, 'J' menjadi angka 10, dan seterusnya.
5. int bubbleSort(int *cards, int n) { ... }: Fungsi ini mengurutkan array kartu menggunakan algoritma Bubble Sort dan mengembalikan jumlah langkah pertukaran yang dilakukan.
  -cards: Pointer ke array kartu.
  -n: Jumlah kartu dalam array.
6. int main() { ... }: Fungsi main() adalah titik masuk utama program.
7. int N;: Mendeklarasikan variabel N untuk menyimpan jumlah kartu.
8. scanf("%d", &N);: Meminta pengguna untuk memasukkan jumlah kartu dan menyimpannya dalam variabel N.
9. int *cards = (int *)malloc(N * sizeof(int));: Membuat array dinamis untuk menyimpan kartu-kartu dengan ukuran sebesar N.
10. char input[3];: Membuat array untuk menyimpan input sementara dari pengguna.
11. for (int i = 0; i < N; i++) { ... }: Loop ini membaca kartu-kartu dari input, mengonversi mereka ke nilai angka, dan menyimpannya dalam array cards.
12. int minSteps = bubbleSort(cards, N);: Memanggil fungsi bubbleSort() untuk mengurutkan kartu dan menyimpan jumlah langkah pertukaran minimal.
13. printf("Jumlah langkah pertukaran: %d\n", minSteps);: Mencetak jumlah langkah pertukaran yang dilakukan selama pengurutan kartu.
14. free(cards);: Membebaskan memori yang dialokasikan untuk array kartu setelah selesai digunakan.
15. return 0; : mengembalikan nilai 0 yang menunjukan bahwa program berhasil dijalankan.

2. Kuda Kobo

Source Code

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Array untuk menyimpan pergerakan kuda
    int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1},
                       {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};

    // Melakukan pengecekan dan menandai posisi yang dapat dicapai oleh kuda
    for (int k = 0; k < 8; k++) {
        int x = i + moves[k][0];
        int y = j + moves[k][1];
        if (x >= 0 && x < size && y >= 0 && y < size) {
            chessBoard[x * size + y] = 1;
        }
    }

    // Menampilkan papan catur setelah simulasi pergerakan kuda
    printf("Papan catur setelah simulasi pergerakan kuda:\n");
    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            printf("%d ", chessBoard[x * size + y]);
        }
        printf("\n");
    }
}

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

    int size = 8;
    int *chessBoard = (int *)malloc(size * size * sizeof(int));
    // Menginisialisasi papan catur dengan nilai 0
    for (int k = 0; k < size * size; k++) {
        chessBoard[k] = 0;
    }

    // Memanggil fungsi koboImaginaryChess untuk mensimulasikan pergerakan
kuda
    koboImaginaryChess(i, j, size, chessBoard);

    free(chessBoard); // Membebaskan memori yang dialokasikan untuk papan
catur
```

```
    return 0;
}
```

Output :

```
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test 04.exe'
3 7
Papan catur setelah simulasi pergerakan kuda:
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
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test 04.exe'
4 5
Papan catur setelah simulasi pergerakan kuda:
0 0 0 0 0 0 0 0
0 0 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
PS C:\Users\Daniel\Downloads\PS1\output>
```

Penjelasan :
1. #include <stdio.h>: Mengimpor file header stdio.h, yang berisi fungsi-fungsi standar input-output untuk operasi masukan dan keluaran.
2. #include <stdlib.h>: Mengimpor file header stdlib.h, yang berisi fungsi-fungsi standar untuk alokasi memori dinamis dan fungsi-fungsi umum lainnya.
3. void koboImaginaryChess(int i, int j, int size, int *chessBoard) { ... }: Deklarasi fungsi koboImaginaryChess yang bertanggung jawab untuk mensimulasikan pergerakan kuda di papan catur.
4. int main() { ... }: Fungsi utama main() dari program.
5. int i, j;: Deklarasi variabel i dan j untuk menyimpan posisi awal kuda.
6. scanf("%d %d", &i, &j);: Meminta pengguna untuk memasukkan posisi awal kuda (koordinat i dan j) dan menyimpannya dalam variabel i dan j.
7. int size = 8;: Deklarasi variabel size untuk menentukan ukuran papan catur, yang dalam kasus ini adalah 8x8.

8. int *chessBoard = (int *)malloc(size * size * sizeof(int));: Mengalokasikan memori untuk papan catur menggunakan malloc() dengan ukuran size * size.
9. for (int k = 0; k < size * size; k++) { ... }: Loop ini menginisialisasi setiap sel di papan catur dengan nilai 0 menggunakan perulangan for.
10. koboImaginaryChess(i, j, size, chessBoard);: Memanggil fungsi koboImaginaryChess untuk mensimulasikan pergerakan kuda di papan catur.
11. free(chessBoard);: Membebaskan memori yang dialokasikan untuk papan catur setelah selesai digunakan.
12. return 0; : mengembalikan nilai 0 yang menunjukan bahwa program berhasil dijalankan