

NAMA : MUCHAMMAD KEVIN ARDIANSYAH

KELAS : IF-03-01

NIM : 1203230096

KARTU

Berikut penjelasan rinci mengenai setiap bagian dari program:

Fungsi minSwaps

Deskripsi: Fungsi ini mengimplementasikan algoritma seleksi sort untuk mengurutkan array kartu dan menghitung jumlah minimum swaps yang diperlukan.

Parameter:

arr[]: Array yang berisi nilai-nilai kartu.

n: Jumlah elemen dalam array kartu.

Algoritma Seleksi Sort:

Iterasi dilakukan sebanyak $n - 1$ kali (dimulai dari 0).

Pada setiap iterasi, dilakukan pencarian nilai terkecil dalam array yang belum diurutkan.

Nilai terkecil tersebut dipertukarkan dengan nilai pada indeks saat ini jika tidak sama dengan indeks saat ini.

Setelah setiap pertukaran, nilai minimum dan nilai yang dipertukarkan ditampilkan.

Output: Fungsi ini mengembalikan jumlah minimum swaps yang diperlukan untuk mengurutkan array kartu.

Fungsi main

Deskripsi: Fungsi utama dari program, bertanggung jawab untuk menerima input pengguna, memanggil fungsi minSwaps, dan menampilkan hasilnya.

Langkah-langkah:

Meminta pengguna untuk memasukkan jumlah kartu.

Memeriksa apakah jumlah kartu valid.

Membuat array untuk menyimpan urutan kartu.

Meminta pengguna untuk memasukkan urutan kartu.

Mengonversi input pengguna menjadi array angka yang sesuai.

Memeriksa apakah jumlah token sesuai dengan jumlah kartu.

Memanggil fungsi minSwaps untuk menghitung jumlah minimum swaps.

Menampilkan jumlah minimum swaps yang diperlukan.

```
Masukkan jumlah kartu: 4
Masukkan urutan kartu (pisahkan dengan spasi): 6 6 9 7
Pengurutan ke-1:
Setelah pertukaran: 6 6 7 9

Jumlah minimum swaps yang diperlukan untuk mengurutkan kartu: 1
```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Fungsi untuk menentukan jumlah minimum swaps yang diperlukan untuk mengurutkan array
6  int minSwaps(int arr[], int n) {
7      int swaps = 0;
8
9      // Loop untuk melakukan seleksi sort dan menghitung swaps
10     for (int i = 0; i < n - 1; i++) {
11         int minIndex = i;
12
13         // Loop untuk mencari nilai minimum dalam array
14         for (int j = i + 1; j < n; j++) {
15             if (arr[j] < arr[minIndex])
16                 minIndex = j;
17         }
18
19         // Jika nilai minimum tidak berada di indeks saat ini, lakukan pertukaran
20         if (minIndex != i) {
21             swaps++;
22
23             // Lakukan pertukaran nilai
24             int temp = arr[i];
25             arr[i] = arr[minIndex];
26             arr[minIndex] = temp;
27
28             // Tampilkan hasil pertukaran
29             printf("Pengurutan ke-%d:\n", swaps);
30             printf("Setelah pertukaran: ");
31             for (int k = 0; k < n; k++) {
32                 // Konversi nilai kartu jika diperlukan
33                 if (arr[k] >= 11 && arr[k] <= 13) {
34                     char card;
35                     switch (arr[k]) {
36                         case 11:
37                             card = 'J';
38                             break;
39                         case 12:
40                             card = 'Q';
41                             break;
42                         case 13:
43                             card = 'K';
44                             break;
45                     }
46                     printf("%c ", card);
47                 } else {
48                     printf("%d ", arr[k]);
49                 }
50             }
51             printf("\n\n");
52         }
53     }
54
55     return swaps;
56 }
57
58 int main() {
59     int n;
60     // Meminta pengguna untuk memasukkan jumlah kartu
61     printf("Masukkan jumlah kartu: ");
62     scanf("%d", &n);
63
64     // Memeriksa apakah jumlah kartu valid
65     if (n <= 0 || n > 100) {
66         printf("Jumlah kartu tidak valid. Program berhenti.\n");
67         return 1;
68     }
69
70     // Array untuk menyimpan urutan kartu
71     int arr[n];
72     // String untuk menyimpan input pengguna
73     char str[3 * n];
74
75     // Meminta pengguna untuk memasukkan urutan kartu
76     printf("Masukkan urutan kartu (pisahkan dengan spasi): ");
77     getchar();
78     fgets(str, sizeof(str), stdin);
79
80     // Tokenisasi input pengguna dan memasukkan nilai ke dalam array
81     char *token = strtok(str, " ");
82     for (int i = 0; i < n && token != NULL; i++) {
83         if (token[0] >= '2' && token[0] <= '9')
84             arr[i] = atoi(token);
85         else
86             arr[i] = (token[0] == 'J') ? 11 : (token[0] == 'Q') ? 12 : (token[0] == 'K') ? 13 : 0;
87
88         token = strtok(NULL, " ");
89     }
90
91     // Memeriksa apakah jumlah token sesuai dengan jumlah kartu
92     if (token != NULL) {
93         printf("Jumlah kartu tidak sesuai. Program berhenti.\n");
94         return 1;
95     }
96
97     // Memanggil fungsi untuk menghitung minimum swaps dan menampilkan hasilnya
98     int swaps = minSwaps(arr, n);
99     printf("Jumlah minimum swaps yang diperlukan untuk mengurutkan kartu: %d\n", swaps);
100
101     return 0;
102 }
103

```

KUDA

Fungsi imaginaryKoboChess

Deskripsi: Fungsi ini menerima posisi baris dan kolom Kobo serta papan catur sebagai parameter, lalu menandai kotak-kotak di mana Kobo bisa bergerak.

Parameter:

row: Posisi baris Kobo.

col: Posisi kolom Kobo.

board[8][8]: Array 2 dimensi yang merepresentasikan papan catur.

Langkah-langkah:

Didefinisikan array steps yang berisi kemungkinan langkah Kobo di atas papan catur.

Iterasi dilakukan untuk setiap kemungkinan langkah.

Di setiap iterasi, posisi baris dan kolom Kobo diubah sesuai dengan langkah yang sedang diuji.

Jika posisi yang baru berada dalam batas papan catur (antara 0 hingga 7 untuk kedua baris dan kolom), kotak tersebut ditandai dengan 1 di papan catur.

Fungsi printChessBoard

Deskripsi: Fungsi ini menerima papan catur sebagai parameter dan mencetaknya ke layar.

Parameter:

board[8][8]: Array 2 dimensi yang merepresentasikan papan catur.

Langkah-langkah:

Iterasi dilakukan untuk setiap kotak di papan catur.

Setiap kotak dicetak ke layar dengan tanda 0 jika tidak diisi oleh Kobo dan 1 jika diisi oleh Kobo.

Fungsi main

Deskripsi: Fungsi utama dari program, bertanggung jawab untuk menerima input posisi Kobo dari pengguna, memanggil fungsi imaginaryKoboChess, dan menampilkan hasilnya.

Langkah-langkah:

Meminta pengguna untuk memasukkan posisi baris dan kolom Kobo.

Membaca posisi baris dan kolom Kobo dari input pengguna.

Inisialisasi papan catur dengan semua kotak bernilai 0.

Memanggil fungsi `imaginaryKoboChess` untuk menandai kotak-kotak yang bisa dijangkau oleh Kobo.

Memanggil fungsi `printChessBoard` untuk mencetak papan catur yang telah dimodifikasi ke layar.

```
1  #include <stdio.h>
2
3  void imaginaryKoboChess(int row, int col, int board[8][8]) {
4
5      int steps[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1},
6                          {2, 1}, {1, 2}, {-1, 2}, {-2, 1}};
7
8
9      for (int k = 0; k < 8; k++) {
10         int newRow = row + steps[k][0];
11         int newCol = col + steps[k][1];
12
13         if (newRow >= 0 && newRow < 8 && newCol >= 0 && newCol < 8) {
14             board[newRow][newCol] = 1;
15         }
16     }
17 }
18
19
20 void printChessBoard(int board[8][8]) {
21     for (int i = 0; i < 8; i++) {
22         for (int j = 0; j < 8; j++) {
23             printf("%d ", board[i][j]);
24         }
25         printf("\n");
26     }
27 }
28
29 int main() {
30     int row, col;
31     scanf("%d %d", &row, &col);
32     int board[8][8] = {{0}};
33     imaginaryKoboChess(row, col, board);
34     printChessBoard(board);
35     return 0;
36 }
37
```