

Nama : Ajax Amstermarstama Januariel

IF03-03

NIM : 1203230091

Kode no.1

```
#include <stdio.h>

void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int minSwaps(int arr[], int n) {
    int i, j, min, min_index, count = 0;

    for (i = 0; i < n-1; i++) {
        min = arr[i];
        min_index = i;
        for (j = i+1; j < n; j++) {
            if (arr[j] < min) {
                min = arr[j];
                min_index = j;
            }
        }

        if (min_index != i) {
            swap(&arr[i], &arr[min_index]);
            count++;
        }
    }

    return count;
}

int main() {
    int n, i, arr[10];

    // Input jumlah kartu dan nilai kartu
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Hitung jumlah minimal pertukaran
    int count = minSwaps(arr, n);

    // Tampilkan urutan kartu setiap terjadi pertukaran
    printf("%d\n", count);

    return 0;
}
```

Penjelasan

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

Kegunaan dari kode di atas adalah untuk mengimpor tiga file header standar yang sering digunakan dalam pemrograman C. Dengan mengimpor header-header ini, pengguna dapat menggunakan fungsi-fungsi standar yang didefinisikan di dalamnya tanpa perlu menulis implementasi dari fungsi-fungsi tersebut secara manual.

```
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
```

Jadi, fungsi swap digunakan untuk menukar nilai dari dua variabel, yang berguna dalam banyak kasus seperti dalam algoritma pengurutan atau dalam implementasi struktur data. Dengan menggunakan fungsi ini, programmer dapat dengan mudah menukar nilai antara dua variabel tanpa perlu menulis ulang kode pertukaran nilai tersebut setiap kali dibutuhkan.

```
int minSwaps(int arr[], int n) {
    int i, j, min, min_index, count = 0;

    for (i = 0; i < n-1; i++) {
        min = arr[i];
        min_index = i;
        for (j = i+1; j < n; j++) {
```

```

        if (arr[j] < min) {
            min = arr[j];
            min_index = j;
        }
    }

    if (min_index != i) {
        swap(&arr[i], &arr[min_index]);
        count++;
    }
}

return count;
}

```

Fungsi minSwaps yang diberikan adalah sebuah fungsi dalam bahasa pemrograman C yang bertugas untuk menghitung jumlah minimum pertukaran yang diperlukan untuk mengurutkan array secara ascending menggunakan algoritma selection sort. Fungsi ini merupakan implementasi dari algoritma selection sort yang efisien dalam mencari nilai minimum dan melakukan pertukaran hanya jika diperlukan. Itu menghasilkan jumlah minimum pertukaran yang diperlukan untuk mengurutkan array.

```

int main() {
    int n, i, arr[10];

    // Input jumlah kartu dan nilai kartu
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Hitung jumlah minimal langkah pertukaran
    int count = minSwaps(arr, n);

    //Tampilan urutan kartu setiap terjadi pertukaran
    printf("%d\n", count);

    return 0;
}

```

Dengan demikian, fungsi main() ini bertanggung jawab untuk mengatur proses input nilai kartu, menghitung jumlah minimal langkah pertukaran, dan menampilkan hasilnya kepada pengguna.

## Output

```

PS C:\Users\User\AppData\Local\Temp> cd "C:\Users\User\AppData\Local\Temp\"; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile }; if ($?) { .\tempCodeRunnerFile }
6
8 3 k q 3 2
4
PS C:\Users\User\AppData\Local\Temp> cd "C:\Users\User\AppData\Local\Temp\"; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile }; if ($?) { .\tempCodeRunnerFile }
5
3 2 8 7 4
2
PS C:\Users\User\AppData\Local\Temp> cd "C:\Users\User\AppData\Local\Temp\"; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile }; if ($?) { .\tempCodeRunnerFile }
6 8 9 7
4
1
PS C:\Users\User\AppData\Local\Temp>

```

Kode no.2

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Array 2D untuk menyimpan papan catur
    int board[size][size];

    // Inisialisasi semua elemen papan catur dengan nilai 0
    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            board[row][col] = 0;
        }
    }

    // Menentukan posisi yang dapat dicapai oleh bidak kuda
    int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
                      {1, -2}, {1, 2}, {2, -1}, {2, 1}};

    // Mengupdate posisi bidak kuda dan menandai posisi yang dapat dicapai
    for (int m = 0; m < 8; m++) {
        int new_i = i + moves[m][0];
        int new_j = j + moves[m][1];

        // Memeriksa apakah posisi baru valid
        if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
            board[new_i][new_j] = 1;
        }
    }

    // Menampilkan papan catur setelah update
    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            printf("%d ", board[row][col]);
        }
        printf("\n");
    }
}

int main() {
    int i, j;
    printf("Masukkan nilai i dan j dipisahkan spasi: ");
    scanf("%d %d", &i, &j);

    // Ukuran papan catur
    int size = 8;

    // Memanggil fungsi koboImaginaryChess
    koboImaginaryChess(i, j, size, NULL);

    return 0;
}
```

## Penjelasan

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

Kode ini adalah program C sederhana yang menggunakan dua header files standar: `stdio.h` dan `stdlib.h`. Ini menunjukkan bahwa program akan menggunakan fungsi-fungsi standar input/output dan fungsi-fungsi dasar yang didefinisikan dalam C.

```
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
```

Fungsi `koboImaginaryChess` memiliki tugas untuk mengisi papan catur imajiner dengan ukuran `size` mulai dari posisi `(i, j)` menggunakan representasi papan catur dalam bentuk array `chessBoard`.

```
    // Array 2D untuk menyimpan papan catur
    int board[size][size];
```

Kode di atas mendeklarasikan sebuah array 2D bernama `board` dengan ukuran `size` kali `size`. Array ini bertujuan untuk menyimpan papan catur dalam representasi matriks.

```
    // Inisialisasi semua elemen papan catur dengan nilai 0
    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            board[row][col] = 0;
        }
    }
```

Kode di atas merupakan sebuah loop bersarang yang digunakan untuk menginisialisasi semua elemen pada papan catur dengan nilai 0. semua sel pada papan catur akan diatur ke nilai awal yang sesuai sebelum operasi atau permainan apa pun dimulai. Ini adalah langkah yang umum dilakukan dalam pemrograman untuk memastikan bahwa nilai-nilai awal sudah ditentukan sebelum digunakan.

```
    // Menentukan posisi yang dapat dicapai oleh bidak kuda
    int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
```

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

Mendefinisikan isi array moves dengan semua kemungkinan pergerakan bidak kuda. Setiap baris merepresentasikan satu pergerakan, dengan elemen pertama menyatakan perubahan baris (delta\_baris) dan elemen kedua menyatakan perubahan kolom (delta\_kolom).

```
// Mengupdate posisi bidak kuda dan menandai posisi yang dapat dicapai
for (int m = 0; m < 8; m++) {
    int new_i = i + moves[m][0];
    int new_j = j + moves[m][1];
```

sebuah loop yang digunakan untuk mengupdate posisi bidak kuda dan menandai posisi yang dapat dicapai di papan catur. pengguna dapat menandai semua posisi yang dapat dicapai oleh bidak kuda dari posisi awal yang diberikan di papan catur.

```
// Memeriksa apakah posisi baru valid
if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
    board[new_i][new_j] = 1;
}
}
```

kode tersebut bertujuan untuk memastikan bahwa posisi baru yang akan diatur di dalam papan permainan berada dalam batas yang valid, sehingga tidak menyebabkan akses di luar indeks atau di luar batas papan.

```
// Menampilkan papan catur setelah update
for (int row = 0; row < size; row++) {
    for (int col = 0; col < size; col++) {
        printf("%d ", board[row][col]);
    }
    printf("\n");
}
}
```

kode tersebut bertanggung jawab untuk menampilkan papan catur setelah dilakukan pembaruan posisi, sehingga pengguna dapat melihat perubahan yang terjadi.

```
int main() {
    int i, j;
    printf("Masukkan nilai i dan j dipisahkan spasi: ");
    scanf("%d %d", &i, &j);

    // Ukuran papan catur
    int size = 8;

    // Memanggil fungsi koboImaginaryChess
    koboImaginaryChess(i, j, size, NULL);

    return 0;
}
```

Dengan demikian, program ini akan meminta pengguna untuk memasukkan koordinat awal posisi bidak kuda, lalu menghasilkan dan menampilkan papan catur imajiner berdasarkan posisi tersebut.

## Output

```
PS C:\Users\User\AppData\Local\Temp> cd "C:\Users\User\AppData\Local\Temp\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile ; if ($?) { .\tempCodeRunnerFile }
Masukkan nilai i dan j dipisahkan spasi: 2 2
0 1 0 0 0 0
1 0 0 1 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 0
PS C:\Users\User\AppData\Local\Temp> cd "C:\Users\User\AppData\Local\Temp\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile ; if ($?) { .\tempCodeRunnerFile }
Masukkan nilai i dan j dipisahkan spasi: 3 7
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
PS C:\Users\User\AppData\Local\Temp> |
```

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