

```

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

int card_value(char card) {
    if (card >= '2' && card <= '9'){ //menentukan
minimal dan maksimal kartu
        return card - '0';
    } else if (card == 'J' || card == 'j'){
//definisi kartu j sebagai 11
        return 11;
    } else if (card == 'Q' || card == 'q'){
//definisi kartu q sebagai 12
        return 12;
    } else if (card == 'K' || card == 'k'){ //definisi
kartu k sebagai 13
        return 13;
    }
    return 0;
}

int bubble_sort(char cards[], int n) {
    int i, j;
    int steps = 0;
    int swapped;
    for (i = 0; i < n - 1; i++) {
        swapped = 0;    // Menetapkan swapped ke 0 di
awal
        for (j = 0; j < n - i - 1; j++) {
            if (card_value(cards[j]) >
card_value(cards[j + 1])) {    // Membandingkan nilai
kartu.
                char temp = cards[j];    // Menyimpan
nilai kartu sementara.

```

```

        cards[j] = cards[j + 1];    // Menukar
nilai kartu.
        cards[j + 1] = temp;    //
Mengembalikan nilai kartu sementara ke posisi
sebelumnya.
        steps++;
        swapped = 1;
        printf("Pertukaran %d: ",
steps);    // Mencetak langkah pertukaran.
        for (int k = 0; k < n; k++) {    //
Mencetak kartu-kartu setelah pertukaran.
            printf("%c ", cards[k]);
        }
            printf("\n");

        if (steps == 5) {    // Memeriksa
apakah sudah dilakukan lima pertukaran.
            printf("Melakukan pertukaran
sebanyak: %d\n", steps);    // Mencetak jumlah
pertukaran.

            return steps;    // Mengembalikan
jumlah pertukaran.
        }
    }
}

    if (swapped == 0)    // Jika tidak ada
pertukaran, berhenti.
        break;
}
    printf("Melakukan pertukaran sebanyak: %d\n",
steps);    // Mencetak jumlah pertukaran.
    return steps;    // Mengembalikan jumlah
pertukaran.
}

```

1. CARDS

```
int main() {
    int n;
    printf("Masukkan jumlah kartu: ");
    scanf("%d", &n);

    char *cards = (char *)malloc(n *
sizeof(char)); // Mengalokasikan memori untuk
kartu-kartu.

    printf("Masukkan nilai kartu (spasi): ");
    for (int i = 0; i < n; i++) { // Mengulangi
proses menerima input untuk setiap kartu.
        scanf(" %c", &cards[i]);
    }
    int steps = bubble_sort(cards, n); // Memanggil
fungsi bubble_sort untuk mengurutkan kartu.
    printf("\nMelakukan pertukaran sebanyak: %d\n",
steps); // Mencetak jumlah pertukaran.

    free(cards); // Membebaskan memori yang
dialokasikan untuk kartu.

    return 0;
}
```

OUTPUT:

```
4
Masukkan nilai kartu (spasi): 6 6 9 7
Pertukaran 1: 6 6 7 9
Melakukan pertukaran sebanyak: 1
```

```
5
Masukkan nilai kartu (spasi): 3 2 8 7 4
Pertukaran 1: 2 3 8 7 4
Pertukaran 2: 2 3 7 8 4
Pertukaran 3: 2 3 7 4 8
Pertukaran 4: 2 3 4 7 8
Melakukan pertukaran sebanyak: 4
```

```
6
Masukkan nilai kartu (spasi): 10 J K Q 3 2
Pertukaran 1: 1 0 J Q K 3
Pertukaran 2: 1 0 J Q 3 K
Pertukaran 3: 1 0 J 3 Q K
Pertukaran 4: 1 0 3 J Q K
Melakukan pertukaran sebanyak: 4
```

```
8
Masukkan nilai kartu (spasi): 9 4 2 J K 8 4 Q
Pertukaran 1: 4 9 2 J K 8 4 Q
Pertukaran 2: 4 2 9 J K 8 4 Q
Pertukaran 3: 4 2 9 J 8 K 4 Q
Pertukaran 4: 4 2 9 J 8 4 K Q
Pertukaran 5: 4 2 9 J 8 4 Q K
Melakukan pertukaran sebanyak: 5
```

2. CHESS

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

void koboImaginaryChess(int i, int j, int size, int
*chessBoard) {
    // fungsi yang menentukan benar atau tidak nya
posisi kuda
    if (i < 0 || j < 0 || i >= size || j >= size) {
        printf("Koordinat tidak valid\n");
        return;
    }

    //semua kemungkinan perpindahan
    int moves[8][2] = {
        {-2, -1}, {-1, -2}, {1, -2}, {2, -1},
        {-2, 1}, {-1, 2}, {1, 2}, {2, 1}
```

```

};

//menyatakan semua kemungkinan diatas lalu
ditandai dengan 1
for (int k = 0; k < 8; k++) {
    int ni = i + moves[k][0];
    int nj = j + moves[k][1]; //
    if (ni >= 0 && nj >= 0 && ni < size && nj <
size) {
        chessBoard[ni * size + nj] = 1;
    }
}
}

int main() {

    int chessBoard[8][8] = {0};

    int i, j;
    printf("Masukkan letak kuda ");
    scanf("%d %d", &i, &j);

    koboImaginaryChess(i, j, 8, (int *)chessBoard);

    //menampilkan papan catur yang telah di perbarui
    printf("Papan catur setelah diperbarui:\n");
    for (int row = 0; row < 8; row++) {
        for (int col = 0; col < 8; col++) {
            printf("%d ", chessBoard[row][col]);
        }
        printf("\n");
    }
}

```

```
    return 0;  
}
```

OUTPUT:

```
Masukkan letak kuda 3 7  
Papan catur setelah diperbarui:  
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
```

```
Masukkan letak kuda 2 2  
Papan catur setelah diperbarui:  
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 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```