

1. Input code :

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

int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

int main() {
    int n, i, j, min_swp = 0;
    char card[100];

    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf(" %c", &card[i]);
    }

    for (i = 0; i < n; i++) {
        int min_idx = i;

        for (j = i + 1; j < n; j++) {
            if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {
                min_idx = j;
            }
        }

        if (min_idx != i) {
            char temp = card[i];
            card[i] = card[min_idx];
            card[min_idx] = temp;

            min_swp++;
        }
    }
}
```

```

printf("%d\n", min_swp);
return 0;
}

```

Hasil Output:

```

PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
4
6 6 9 7
1
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
5
3 2 8 7 4
2
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
6
10 J K Q 3 2
4
PS C:\AlPro\ASD>

```

Penjelasan:

```

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

```

Hal diatas ialah header yang digunakan untuk fungsi-fungsi yang digunakan di dalam program.

```

int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

```

Fungsi tersebut adalah fungsi nilai kartu untuk memanggil karakter nilai kartu atau mengembalikan sebuah numerik dari kartu tersebut. 'J', 'Q', 'K' dan '1' masing masing diwakili oleh nilai numerik 11, 12, 13 dan 10. Sementara itu kartu-kartu numerik tersebut diwakili oleh nilai numerik yang sama dengan cara pengurangan karakter. Misalnya, '2' akan diwakili oleh 2, '3' akan diwakili oleh 3 dan seterusnya.

```

int main() {
    int n, i, j, min_swp = 0;
    char card[100];

    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf(" %c", &card[i]);
    }

    for (i = 0; i < n; i++) {
        int min_idx = i;

        for (j = i + 1; j < n; j++) {
            if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {
                min_idx = j;
            }
        }

        if (min_idx != i) {
            char temp = card[i];
            card[i] = card[min_idx];
            card[min_idx] = temp;

            min_swp++;
        }
    }

    printf("%d\n", min_swp);
    return 0;
}

```

Didalam fungsi main tersebut terdapat program yang digunakan untuk membaca kartu yang akan dimasukkan. Kemudian, program tersebut akan membaca karakter-karakter yang mewakili kartu tersebut dan akan menyimpannya di dalam sebuah array. Setelah itu program akan melakukan seleksi dengan cara mencari kartu dengan nilai numerik terkecil dan menukar kartu tersebut dengan kartu di index saat ini. Jumlah pertukaran tersebut disimpan dalam variabel 'min_swp' dan dicetak setelah pengurutan selesai.

Jadi kode diatas merupakan kode yang digunakan untuk mengurutkan kartu-kartu berdasarkan nilai numeriknya, dan kemudian mencetak jumlah minimum pertukaran yang dilakukan selama proses pengurutan.

Bonus

Input kode:

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

// Fungsi untuk mendapatkan nilai numerik dari kartu
int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

int main() {
    int n, i, j, min_swp = 0;
    char card[100];

    scanf("%d", &n);

    // Memeriksa apakah ada kartu yang dimasukkan
    if (n > 0) {
        for (i = 0; i < n; i++) {
            scanf(" %c", &card[i]);
        }

        for (i = 0; i < n; i++) {
            int min_idx = i;

            for (j = i + 1; j < n; j++) {
                if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {
                    min_idx = j;
                }
            }

            if (min_idx != i) {
                char temp = card[i];
                card[i] = card[min_idx];
                card[min_idx] = temp;
            }
        }
    }
}
```

```

        // Menampilkan urutan kartu setiap pertukaran
        printf("Pertukaran ke-%d: ", min_swp + 1);
        for (int k = 0; k < n; k++) {
            printf("%c ", card[k]);
        }
        printf("\n");

        min_swp++;
    }
}

printf("Jumlah minimum pertukaran: %d\n", min_swp);
return 0;
}

```

Hasil Output:

```

PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalpraktikum }
8
9 4 2 J K 8 4 Q
Pertukaran ke-1: 2 4 9 J K 8 4 Q
Pertukaran ke-2: 2 4 4 J K 8 9 Q
Pertukaran ke-3: 2 4 4 8 K J 9 Q
Pertukaran ke-4: 2 4 4 8 9 J K Q
Pertukaran ke-5: 2 4 4 8 9 J Q K
Jumlah minimum pertukaran: 5
PS C:\AlPro\ASD>

```

Penjelasan:

```

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

// Fungsi untuk mendapatkan nilai numerik dari kartu
int nilai_kartu(char kartu) {
    switch(kartu) {
        case 'J':
            return 11;
        case 'Q':
            return 12;
        case 'K':
            return 13;
        case '1':
            return 10;
        default:
            return kartu - '0';
    }
}

```

```
}  
}
```

Fungsi ini mendapatkan nilai numerik yang sesuai dengan setiap kartu yang diberikan. Jika kartu 'J', 'Q', 'K', atau '1' maka fungsi ini mengembalikan nilai yang ditentukan.

```
int main() {  
    int n, i, j, min_swp = 0;  
    char card[100];  
  
    scanf("%d", &n);  
  
    // Memeriksa apakah ada kartu yang dimasukkan  
    if (n > 0) {  
        for (i = 0; i < n; i++) {  
            scanf(" %c", &card[i]);  
        }  
  
        for (i = 0; i < n; i++) {  
            int min_idx = i;  
  
            for (j = i + 1; j < n; j++) {  
                if (nilai_kartu(card[j]) < nilai_kartu(card[min_idx])) {  
                    min_idx = j;  
                }  
            }  
  
            if (min_idx != i) {  
                char temp = card[i];  
                card[i] = card[min_idx];  
                card[min_idx] = temp;  
  
                // Menampilkan urutan kartu setiap pertukaran  
                printf("Pertukaran ke-%d: ", min_swp + 1);  
                for (int k = 0; k < n; k++) {  
                    printf("%c ", card[k]);  
                }  
                printf("\n");  
  
                min_swp++;  
            }  
        }  
    }  
  
    printf("Jumlah minimum pertukaran: %d\n", min_swp);  
    return 0;  
}
```

Didalam fungsi main, program membaca input 'n' yang merupakan jumlah kartu. Program kemudian memasukkan kartu-kartu ke dalam array. Selanjutnya program melakukan pengurutan menggunakan algoritma menyeleksi. Ia mencari nilai minimum dan menukar elemen-elemen jika diperlukan. Setiap terjadi pertukaran, program mencetak urutan kartu baru dan juga program akan mencetak jumlah minimum pertukaran yang diperlukan untuk mengurutkan kartu-kartu tersebut.

Jadi program diatas merupakan implementasi dari algoritma pengurutan seleksi(selection sort) yang mengurutkan array of char yang merepresentasikan kartu-kartu. Setiap kartu memiliki nilai yang diasosiasikan dan pengurutan dilakukan berdasarkan nilai numerik tersebut.

2. Input Code:

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

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

// Fungsi untuk menandai posisi yang dapat dicapai oleh kuda
void markAccessiblePositions(int i, int j, int *chessBoard) {
    // Daftar langkah yang mungkin dilakukan oleh kuda
    int moves[8][2] = {
        {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
        {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
    };

    // Memeriksa setiap langkah yang mungkin
    for (int k = 0; k < 8; k++) {
        int x = i + moves[k][0];
        int y = j + moves[k][1];
        if (isValidPosition(x, y)) {
            chessBoard[x * 8 + y] = 1; // Menandai posisi yang dapat dicapai
        }
    }
}

// Fungsi untuk mencetak papan catur
void printChessBoard(int *chessBoard) {
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            printf("%d ", chessBoard[i * 8 + j]);
        }
        printf("\n");
    }
}
```

```

// Fungsi utama
void koboImaginaryChess(int i, int j, int *chessBoard) {
    // Mengosongkan papan catur
    for (int k = 0; k < 64; k++) {
        chessBoard[k] = 0;
    }

    // Menandai posisi yang dapat dicapai oleh kuda
    markAccessiblePositions(i, j, chessBoard);
}

int main() {
    int i, j;
    int chessBoard[64] = {0}; // Inisialisasi papan catur dengan nilai 0

    // Membaca posisi kuda
    scanf("%d %d", &i, &j);

    // Memeriksa dan menandai posisi yang dapat dicapai oleh kuda
    koboImaginaryChess(i, j, chessBoard);

    // Mencetak papan catur dengan posisi yang dapat dicapai oleh kuda
    printChessBoard(chessBoard);

    return 0;
}

```

Hasil Ouput:

```

PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalprakti
kum }
2 2
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
PS C:\AlPro\ASD> cd "c:\AlPro\ASD\" ; if ($?) { gcc soalpraktikum.c -o soalpraktikum } ; if ($?) { .\soalprakti
kum }
3 7
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:\AlPro\ASD>

```


Penjelasan:

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

Fungsi ini mengembalikan nilai 1 jika posisi '(x, y)' valid di papan catur 8x8, dan 0 jika tidak.

```
// Fungsi untuk menandai posisi yang dapat dicapai oleh kuda
void markAccessiblePositions(int i, int j, int *chessBoard) {
    // Daftar langkah yang mungkin dilakukan oleh kuda
    int moves[8][2] = {
        {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
        {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
    };

    // Memeriksa setiap langkah yang mungkin
    for (int k = 0; k < 8; k++) {
        int x = i + moves[k][0];
        int y = j + moves[k][1];
        if (isValidPosition(x, y)) {
            chessBoard[x * 8 + y] = 1; // Menandai posisi yang dapat dicapai
        }
    }
}
```

Fungsi ini menandai posisi yang dapat dicapai oleh kuda pada papan catur dengan menandai elemen-elemen yang sesuai dalam array.

```
// Fungsi untuk mencetak papan catur
void printChessBoard(int *chessBoard) {
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            printf("%d ", chessBoard[i * 8 + j]);
        }
        printf("\n");
    }
}
```

Fungsi ini mencetak papan catur dengan posisi yang telah ditandai.

```
// Fungsi utama
void koboImaginaryChess(int i, int j, int *chessBoard) {
    // Mengosongkan papan catur
    for (int k = 0; k < 64; k++) {
        chessBoard[k] = 0;
    }
}
```

```
    // Menandai posisi yang dapat dicapai oleh kuda  
    markAccessiblePositions(i, j, chessBoard);  
}
```

Fungsi ini merupakan fungsi utama yang digunakan untuk menandai posisi-posisi yang dapat oleh kuda pada catur.

```
int main() {  
    int i, j;  
    int chessBoard[64] = {0}; // Inisialisasi papan catur dengan nilai 0  
  
    // Membaca posisi kuda  
    scanf("%d %d", &i, &j);  
  
    // Memeriksa dan menandai posisi yang dapat dicapai oleh kuda  
    koboImaginaryChess(i, j, chessBoard);  
  
    // Mencetak papan catur dengan posisi yang dapat dicapai oleh kuda  
    printChessBoard(chessBoard);  
  
    return 0;  
}
```

Fungsi main ini merupakan fungsi utama program yang bertanggung jawab untuk meminta input.

Jadi, keseluruhan program ini menerima input posisi kuda, menandai posisi-posisi yang dapat dicapai oleh kuda pada papan catur, dan mencetak papan catur tersebut dengan posisi-posisi yang telah ditandai.