**NAMA LENGKAP:MICHAEL VALENTINO SOGEN**

**KELAS :IF 03-02**

**NIM :1203230099**

**MATKUL :ALGORITMA DAN STRUKTUR DATA(Praktikum)**

## CODE PROGRAM

1. **SOURCE CODE**

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

int get_card_value(char card) {
    if (card == 'J') return 11;
    else if (card == 'Q') return 12;
    else if (card == 'K') return 13;
    else if (card == '1') return 10;
    else return card - '0';
}

int main() {
    int n, i, j, min_swaps = 0;
    char cards[100];
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        scanf(" %c", &cards[i]);
    }
    for (i = 0; i < n; i++) {
        int min_idx = i;
        for (j = i + 1; j < n; j++) {
            if (get_card_value(cards[j]) < get_card_value(cards[min_idx])) {
                min_idx = j;
            }
        }
        if (min_idx != i) {
            char temp = cards[i];
            cards[i] = cards[min_idx];
            cards[min_idx] = temp;
            min_swaps++;
        }
    }
    printf("%d\n", min_swaps);
    return 0;
}
```

2. **SOURCE CODE**

```
#include <stdio.h>

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Menginisialisasi semua elemen array dengan nilai 0
    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            *(chessBoard + x * size + y) = 0;
        }
    }

    // Menghitung kemungkinan posisi yang dapat dilalui oleh bidak kuda
    int moves[8][2] = { {-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2},
{1, -2}, {1, 2} };
    for (int k = 0; k < 8; k++) {
        int newX = i + moves[k][0];
        int newY = j + moves[k][1];
        if (newX >= 0 && newX < size && newY >= 0 && newY < size) {
            *(chessBoard + newX * size + newY) = 1;
        }
    }

    // Menampilkan output array
    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 chessBoard[8][8];
    koboImaginaryChess(i, j, 8, (int *)chessBoard);
    return 0;
}
```