

```

1  #include <stdio.h>
2
3  int main() {
4      // Menentukan jumlah kartu.
5      int N;
6      scanf("%d", &N);
7
8      // Membaca nilai kartu.
9      int cards[N];
10     for (int i = 0; i < N; i++) {
11         char card;
12         //mendeklarasikan variabel card
13         scanf(" %c", &card);
14         if (card >= '1' && card <= '9') {
15             //memeriksa apakah karakter input mewakili nilai kartu antara '1' hingga '9'
16             cards[i] = card - '0';
17             //mengonversi karakter yang mewakili nilai kartu
18         } else if (card == 'J') {
19             cards[i] = 11;
20             //Jika karakter input adalah 'J', maka kartu tersebut dianggap bernilai 11.
21         } else if (card == 'Q') {
22             cards[i] = 12;
23             //Jika karakter input adalah 'Q', maka kartu tersebut dianggap bernilai 12.
24         } else if (card == 'K') {
25             cards[i] = 13;
26             //Jika karakter input adalah 'K', maka kartu tersebut dianggap bernilai 13.
27         }
28     }
29
30     //Menghitung jumlah minimal langkah pertukaran.
31     int swaps = 0;
32     //deklarasi dan inisialisasi variabel swaps
33     for (int i = 0; i < N; i++) {
34         //Menemukan kartu minimum di sisa kartu.
35         int min_idx = i;
36         //mendeklarasikan dan menginisialisasi variabel min_idx
37         for (int j = i + 1; j < N; j++) {
38             //mencari kartu minimum di sisa kartu yang belum diurutkan
39             if (cards[j] < cards[min_idx]) {
40                 //digunakan untuk membandingkan nilai kartu
41                 min_idx = j;
42             }
43         }
44
45         // Tukar kartu minimum dengan kartu di posisi saat ini.
46         if (min_idx != i) {
47             //memeriksa apakah min_idx tidak sama dengan i
48             int temp = cards[i];
49             // menyimpan nilai kartu pada indeks i
50             cards[i] = cards[min_idx];
51             // mengganti nilai kartu pada indeks i dengan nilai kartu pada min_idx
52             cards[min_idx] = temp;
53             //mengganti nilai kartu pada min_idx dengan nilai yang disimpan dalam variabel sementara temp
54             swaps++;
55             //pertukaran posisi, jumlah pertukaran (swaps) akan ditambah satu
56         }
57     }
58
59     // Mencetak output.
60     printf("%d\n", swaps);
61
62     return 0;
63 }
64

```

```
PS C:\Users\Rimba BAW\Documents\arrysy\output> & .\'kartu.exe'  
5  
3 2 8 7 4  
2  
PS C:\Users\Rimba BAW\Documents\arrysy\output> & .\'kartu.exe'  
6  
10 j k q 3 2  
4  
PS C:\Users\Rimba BAW\Documents\arrysy\output> & .\'kartu.exe'  
4  
6 6 9 7  
1  
PS C:\Users\Rimba BAW\Documents\arrysy\output> █
```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
5      // Langkah yang mungkin dilakukan oleh bidak kuda
6      int moves[8][2] = { //merepresentasikan langkah-langkah yang mungkin dilakukan oleh bidak kuda
7          {-2, -1}, {-2, 1},
8          //Dua langkah ke atas dan satu langkah ke kiri, Dua langkah ke atas dan satu langkah ke kanan
9          {-1, -2}, {-1, 2},
10         //Satu langkah ke atas dan dua langkah ke kiri, Satu langkah ke atas dan dua langkah ke kanan
11         {1, -2}, {1, 2},
12         //Satu langkah ke bawah dan dua langkah ke kiri, Satu langkah ke bawah dan dua langkah ke kanan
13         {2, -1}, {2, 1}
14         //Dua langkah ke bawah dan satu langkah ke kiri, Dua langkah ke bawah dan satu langkah ke kanan
15     };
16
17     // Melakukan simulasi untuk setiap kemungkinan langkah
18     for (int k = 0; k < 8; k++) {
19         //loop yang akan berjalan sebanyak 8 kali
20         int new_i = i + moves[k][0];
21         //menghitung posisi baris baru
22         int new_j = j + moves[k][1];
23         //menghitung posisi kolom baru
24
25
26         if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
27             // Memastikan posisi baru berada dalam batas papan catur
28             chessBoard[new_i * size + new_j] = 1;
29             // Menandai posisi yang dapat dicapai oleh bidak kuda
30         }
31     }
32 }
33
34 int main() {
35     // Membuat array 2D berukuran 8x8 untuk papan catur
36     int size = 8;
37     //ukuran papan catur
38     int *chessBoard = (int *)malloc(size * size * sizeof(int));
39     //mengalokasikan memori dinamis untuk array 2D merepresentasikan papan catur
40
41     // Inisialisasi papan catur dengan nilai 0
42     for (int i = 0; i < size; i++) {
43         for (int j = 0; j < size; j++) {
44             chessBoard[i * size + j] = 0;
45             //mengatur nilai dari elemen papan catur
46         }
47     }
48
49     // Input posisi bidak kuda
50     int i, j;
51     scanf("%d %d", &i, &j);
52
53     koboImaginaryChess(i, j, size, chessBoard);
54     // Memanggil fungsi untuk mensimulasikan langkah bidak kuda
55
56     // Menampilkan hasil papan catur setelah simulasi
57     for (int i = 0; i < size; i++) {
58         for (int j = 0; j < size; j++) {
59             printf("%d ", chessBoard[i * size + j]);
60             // mencetak nilai dari setiap elemen di papan catur.
61         }
62         printf("\n");
63     }
64
65     free(chessBoard);
66     // Menghapus array dinamis
67

```

```
PS C:\Users\Rimba BAW\Documents\arrysy> cd 'c:\Users\Rimba BAW\Documents\arrysy\output'
```

```
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:\Users\Rimba BAW\Documents\arrysy\output> & .\'array.exe'
```

```
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:\Users\Rimba BAW\Documents\arrysy\output> █
```