

Nama: Muhammad Muhyi Akbar

NM: 1203230090

IF 02

Source code nomer 1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  void swap(char *a, char *b) {
6      char temp = *a;
7      *a = *b;
8      *b = temp;
9  }
10
11 void printArray(char arr[], int n) {
12     for (int i = 0; i < n; i++) {
13         printf("%c ", arr[i]);
14     }
15     printf("\n");
16 }
17
18 int minSteps(char arr[], int n) {
19     int steps = 0;
20     for (int i = 0; i < n - 1; i++) {
21         for (int j = 0; j < n - i - 1; j++) {
22             if (arr[j] > arr[j + 1]) {
23                 swap(&arr[j], &arr[j + 1]);
24                 steps++;
25                 printf("\nPertukaran %d: ", steps);
26                 printArray(arr, n);
27             }
28         }
29     }
30     return steps;
31 }
32
33 int main() {
34     int n;
35     scanf("%d", &n);
36
37     char arr[n];
38     for (int i = 0; i < n; i++) {
39         scanf(" %c", &arr[i]);
40     }
41
42     int steps = minSteps(arr, n);
43     printf("\n%d\n", steps);
44
45     return 0;
46 }
```

Penjelasan source code diatas:

Fungsi swap: Fungsi ini bertugas untuk menukar nilai dua karakter yang diberikan.

Fungsi printArray: Fungsi ini digunakan untuk mencetak elemen-elemen array.

Fungsi minSteps: Fungsi ini melakukan Bubble Sort pada array dan menghitung jumlah langkah (pertukaran) yang diperlukan. Dalam setiap iterasi, jika elemen ke-j lebih besar dari elemen ke-(j+1), maka dilakukan pertukaran dan jumlah langkah ditambah satu.

Fungsi main:

Membaca jumlah elemen array dari input.

Membaca karakter-karakter array.

Memanggil fungsi minSteps untuk mengurutkan array dan menghitung jumlah langkah.

Mencetak jumlah langkah yang diperlukan untuk mengurutkan array.

Output:

```
PS D:\VsCode\.vscode> cd "d:\VsCode\.vscode\" ; if ($?) { gcc kartucatur.c -o kartucatur } ; if ($?) { .\kartucatur }  
5  
3 2 5 6 1  
Pertukaran 1: 2 3 5 6 1  
Pertukaran 2: 2 3 5 1 6  
Pertukaran 3: 2 3 1 5 6  
Pertukaran 4: 2 1 3 5 6  
Pertukaran 5: 1 2 3 5 6  
5  
PS D:\VsCode\.vscode>
```

Source code nomer 2

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
5
6      for (int row = 0; row < size; row++) {
7          for (int col = 0; col < size; col++) {
8              *(chessBoard + row * size + col) = 0;
9          }
10     }
11
12
13     int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
14                       {1, -2}, {1, 2}, {2, -1}, {2, 1}};
15
16
17     for (int k = 0; k < 8; k++) {
18         int newRow = i + moves[k][0];
19         int newCol = j + moves[k][1];
20
21
22         if (newRow >= 0 && newRow < size && newCol >= 0 && newCol < size) {
23             *(chessBoard + newRow * size + newCol) = 1;
24         }
25     }
26
27
28     for (int row = 0; row < size; row++) {
29         for (int col = 0; col < size; col++) {
30             printf("%d", *(chessBoard + row * size + col));
31             if (col < size - 1) {
32                 printf(" ");
33             }
34         }
35         printf("\n");
36     }
37 }
38
39 int main() {
40     int i, j;
41     scanf("%d %d", &i, &j);
42
43
44     int size = 8;
45
46
47     int *chessBoard = (int *)malloc(size * size * sizeof(int));
48
49     koboImaginaryChess(i, j, size, chessBoard);
50
51     free(chessBoard);
52
53     return 0;
54 }
```

Penjelasan Source code diatas:

Fungsi koboImaginaryChess:

Fungsi ini menerima empat parameter: i, j (koordinat awal), size (ukuran papan catur), dan chessBoard (papan catur).

Pertama-tama, papan catur diinisialisasi dengan 0 menggunakan nested loop.

Array moves menyimpan kemungkinan gerakan kuda dalam catur.

Untuk setiap gerakan yang mungkin, fungsi mengecek apakah koordinat baru berada dalam batas papan catur. Jika iya, kotak tersebut ditandai sebagai "1" pada papan catur.

Setelah semua kemungkinan gerakan kuda ditandai, papan catur dicetak.

Fungsi main:

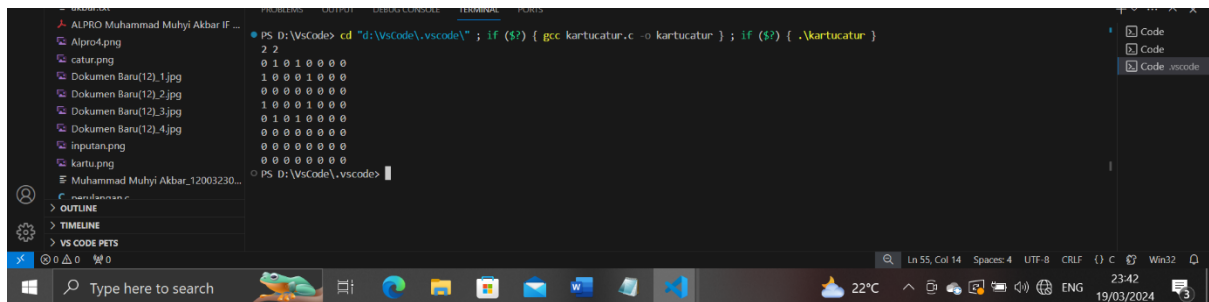
Membaca input koordinat awal (i dan j) dari pengguna.

Membuat papan catur dengan ukuran 8x8 menggunakan alokasi memori dinamis.

Memanggil fungsi koboImaginaryChess dengan parameter yang sesuai.

Membebaskan memori yang dialokasikan untuk papan catur.

Output:



```
PS D:\VsCode> cd "d:\VsCode\"; if ($?) { gcc kartatur.c -o kartatur }; if ($?) { .\kartatur }
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 D:\VsCode>
```