

Nama : Muhammad Naufal Ramadhani

Nim : 1203230036

Kelas : IF 03-03

1. SS source code soal nomer 1

```
C: ltaad.c x
C: ltaad.c > @mindsteps(char [], int)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Fungsi untuk menukar dua elemen dalam array
6  void swap(char *a, char *b) {
7      char temp = *a;
8      *a = *b;
9      *b = temp;
10 }
11
12 // Fungsi untuk mencetak array
13 void printArray(char arr[], int n) {
14     for (int i = 0; i < n; i++) {
15         printf("%c ", arr[i]);
16     }
17     printf("\n");
18 }
19
20 // Fungsi untuk menghitung jumlah langkah minimum
21 int minSteps(char arr[], int n) {
22     int steps = 0;
23     for (int i = 0; i < n - 1; i++) {
24         for (int j = 0; j < n - i - 1; j++) {
25             if (arr[j] > arr[j + 1]) {
26                 swap(&arr[j], &arr[j + 1]);
27                 steps++;
28                 printf("\nPertukaran %d: ", steps);
29                 printArray(arr, n); // Tampilkan urutan kartu setiap terjadi pertukaran
30             }
31         }
32     }
33     return steps;
34 }
35
36 int main() {
37     int n;
38     scanf("%d", &n);
39
40     char arr[n];
41     for (int i = 0; i < n; i++) {
42         scanf(" %c", &arr[i]);
43     }
44
45     int steps = minSteps(arr, n);
46     printf("\n%d\n", steps);
47
48     return 0;
49 }
50
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\K95\OneDrive\Documents\kpa1\IP3.c> cd "C:\Users\K95\OneDrive\Documents\kpa1\IP3.c" ; if ($?) { gcc ltaad.c -o ltaad } ; if ($?) { .\ltaad }
6
50 3 K Q 3 2

Pertukaran 1: 0 1 3 K Q 3
Pertukaran 2: 0 1 3 K 3 Q
Pertukaran 3: 0 1 3 3 K Q

4
PS C:\Users\K95\OneDrive\Documents\kpa1\IP3.c> cd "C:\Users\K95\OneDrive\Documents\kpa1\IP3.c" ; if ($?) { gcc ltaad.c -o ltaad } ; if ($?) { .\ltaad }
4
6 6 9 7
o Pertukaran 1: 6 6 7 9

1
```

Penjelasan program :

1. `#include <stdio.h>`: Mendukung fungsi input-output standar.
2. `#include <stdlib.h>`: Mendukung fungsi-fungsi dasar, seperti alokasi memori.
3. `#include <string.h>`: Mendukung fungsi-fungsi pemrosesan string.
4. `void swap(char *a, char *b) { ... }`: Mendefinisikan fungsi swap yang bertujuan untuk menukar dua elemen dalam array. Fungsi ini menggunakan parameter pointer untuk menerima alamat dua elemen yang akan ditukar.
5. `void printArray(char arr[], int n) { ... }`: Mendefinisikan fungsi printArray yang bertujuan untuk mencetak isi array. Fungsi ini menerima array karakter dan panjangnya sebagai argumen.
6. `int minSteps(char arr[], int n) { ... }`: Mendefinisikan fungsi minSteps yang bertujuan untuk mengurutkan array dengan algoritma bubble sort dan menghitung jumlah langkah minimum yang diperlukan. Fungsi ini mengembalikan jumlah langkah yang dilakukan.
7. `int main() { ... }`: Memulai fungsi utama dari program.
8. `int n`:: Mendeklarasikan variabel `n` yang akan digunakan untuk menyimpan jumlah kartu.
9. `scanf("%d", &n)`:: Membaca jumlah kartu dari input.
10. `char arr[n]`:: Mendeklarasikan array `arr` dengan ukuran `n` yang akan menyimpan nilai-nilai kartu.
11. `for (int i = 0; i < n; i++) { ... }`: Melakukan iterasi untuk membaca nilai kartu satu per satu dan menyimpannya dalam array `arr`.
12. `scanf("%c", &arr[i])`:: Membaca nilai kartu dari input dan menyimpannya dalam array `arr`.
13. `int steps = minSteps(arr, n)`:: Memanggil fungsi `minSteps` untuk mengurutkan array dan menghitung jumlah langkah minimum yang diperlukan. Hasilnya disimpan dalam variabel `steps`.
14. `printf("\n%d\n", steps)`:: Mencetak jumlah langkah minimum yang diperlukan untuk mengurutkan kartu.
15. `return 0`:: Mengakhiri eksekusi program dengan status keluaran yang sukses.

2. SS Source Code soal nomor 2

```
C:latand2 > @ main
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
5     // Inisialisasi semua nilai pada chessBoard dengan 0
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    // Daftar langkah yang mungkin dilakukan oleh bidak kuda
13    int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
14                      {1, -2}, {1, 2}, {2, -1}, {2, 1}};
15
16    // Memperbarui nilai chessBoard untuk setiap posisi yang mungkin dicapai
17    for (int k = 0; k < 8; k++) {
18        int rowCol = i + moves[k][0];
19        int resCol = j + moves[k][1];
20
21        // Memeriksa apakah posisi baru valid
22        if (rowCol >= 0 && rowCol < size && resCol >= 0 && resCol < size) {
23            *(chessBoard + rowCol * size + resCol) = 1;
24        }
25    }
26
27    // Mencetak hasil chessBoard
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     // Ukuran papan catur
44     int size = 8;
45
46     // Membuat array 2D untuk papan catur
47     int *chessBoard = (int *)malloc(size * size * sizeof(int));
48
49     // Memanggil fungsi untuk mensimulasikan gerakan kuda
50     koboImaginaryChess(i, j, size, chessBoard);
51
52     // Menghapus array yang dialokasikan secara dinamis
53     free(chessBoard);
54
55     return 0;
56 }
57
```

```
PS C:\Users\AGUS\OneDrive\Documents\Kopai\TP3.c> cd "C:\Users\AGUS\OneDrive\Documents\Kopai\TP3.c\"; if ($?) { gcc latand2.c -o latand2 }; if ($?) { .\latand2 }
2 2
0 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 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\AGUS\OneDrive\Documents\Kopai\TP3.c> cd "C:\Users\AGUS\OneDrive\Documents\Kopai\TP3.c\"; if ($?) { gcc latand2.c -o latand2 }; if ($?) { .\latand2 }
3 7
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 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 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

PS C:\Users\AGUS\OneDrive\Documents\Kopai\TP3.c>
```

Penjelasan Program :

Program berfungsi untuk mensimulasikan gerakan kuda pada papan catur berukuran 8x8. Alur programnya :

1. Membaca koordinat baris dan kolom dari input.
2. Membuat array 2D yang merepresentasikan papan catur dengan ukuran 8x8.
3. Memanggil fungsi koboImaginaryChess untuk mengisi papan catur dengan nilai yang menunjukkan posisi-posisi yang dapat dicapai oleh kuda dari posisi awal yang diberikan.
4. Setelah selesai, membebaskan memori yang telah dialokasikan secara dinamis untuk array papan catur.
5. Mengakhiri eksekusi program.