Nama : Henri Kurniawan Candra

Kelas : IF-03-02

NIM : 1203230086

Source Code

No. 1

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

void swap(char *a, char *b) {
    char temp = *a;
    *a = *b;
    *b = temp;
}


void printArray(char arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%c ", arr[i]);
    }
    printf("\n");
}


int minSteps(char arr[], int n) {
    int steps = 0;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                swap(&arr[j], &arr[j + 1]);
                steps++;
                printf("\nPertukaran %d: ", steps);
                printArray(arr, n);
            }
        }
    }
    return steps;
}

int main() {
    int n;
    scanf("%d", &n);
```

```
    char arr[n];
    for (int i = 0; i < n; i++) {
        scanf(" %c", &arr[i]);
    }

    int steps = minSteps(arr, n);
    printf("\n%d\n", steps);

    return 0;
}
```

Penjelasan :

Program tersebut adalah sebuah implementasi dari algoritma bubble sort

Fungsi swap digunakan untuk menukar nilai antara dua karakter.
Fungsi printArray digunakan untuk mencetak isi dari array karakter.
Fungsi minSteps mengimplementasikan algoritma bubble sort untuk mengurutkan array karakter dalam urutan naik, dan menghitung jumlah langkah (pertukaran) yang diperlukan.

Di dalam **fungsi main**:
Program menerima input berupa sebuah bilangan bulat n menggunakan scanf.
Array arr dengan ukuran n karakter dideklarasikan dan diisi dengan karakter-karakter yang diinputkan menggunakan loop dan scanf.
Fungsi minSteps dipanggil untuk mengurutkan array dan menghitung jumlah langkah yang diperlukan.
Jumlah langkah yang diperlukan untuk mengurutkan array kemudian dicetak.

Output :

```
PS C:\HENRI\Campus\Project semester 2\Alpro\TugasASD> .\sort10jqk.exe
3
8 5 1

Pertukaran 1: 5 8 1

Pertukaran 2: 5 1 8

Pertukaran 3: 1 5 8

3
```

No. 2

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

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {

    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            *(chessBoard + row * size + col) = 0;
        }
    }


    int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2},
                       {1, -2}, {1, 2}, {2, -1}, {2, 1}};


    for (int k = 0; k < 8; k++) {
        int newRow = i + moves[k][0];
        int newCol = j + moves[k][1];


        if (newRow >= 0 && newRow < size && newCol >= 0 && newCol < size) {
            *(chessBoard + newRow * size + newCol) = 1;
        }
    }


    for (int row = 0; row < size; row++) {
        for (int col = 0; col < size; col++) {
            printf("%d", *(chessBoard + row * size + col));
            if (col < size - 1) {
                printf(" ");
            }
        }
        printf("\n");
    }
}

int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int size = 8;
    int *chessBoard = (int *)malloc(size * size * sizeof(int));


    koboImaginaryChess(i, j, size, chessBoard);
```

```
    free(chessBoard);
    return 0;
}
```

Penjelasan :

Fungsi koboImaginaryChess untuk membuat papan catur imaginasi berukuran 8x8 dan menandai sel-sel yang bisa dijangkau oleh kuda pada posisi tertentu (i, j) di papan catur tersebut.

Fungsi koboImaginaryChess memiliki parameter sebagai berikut:

i dan j: posisi awal kuda pada papan catur.
size: ukuran papan catur (dalam hal ini, ukurannya adalah 8x8).
*chessBoard: pointer ke papan catur yang akan diisi dengan informasi langkah-langkah kuda.

Pada awal fungsi, semua sel pada papan catur diinisialisasi dengan nilai 0.
Array moves digunakan untuk menyimpan langkah-langkah yang mungkin dilakukan oleh kuda.

Selanjutnya, dilakukan pengecekan untuk setiap kemungkinan langkah kuda. Jika langkah tersebut valid (tidak keluar dari batas papan catur), sel pada posisi tersebut akan diisi dengan nilai 1.

Setelah semua kemungkinan langkah kuda telah ditandai pada papan catur, papan catur tersebut akan dicetak ke layar dengan menampilkan nilai-nilai yang menunjukkan apakah suatu sel bisa dijangkau oleh kuda (nilai 1) atau tidak (nilai 0).

Di dalam fungsi main, program membaca input posisi awal kuda (i dan j) dari pengguna menggunakan scanf.
Selanjutnya, dilakukan alokasi memori dinamis untuk papan catur menggunakan malloc.
Fungsi koboImaginaryChess dipanggil dengan parameter yang sesuai, yaitu posisi awal kuda dan papan catur yang sudah dialokasi.
Terakhir, memori yang dialokasikan untuk papan catur dibebaskan menggunakan free.

Output :

```
PS C:\HENRI\Campus\Project semester 2\Alpro\TugasASD> .\papancatur.exe
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:\HENRI\Campus\Project semester 2\Alpro\TugasASD>
```