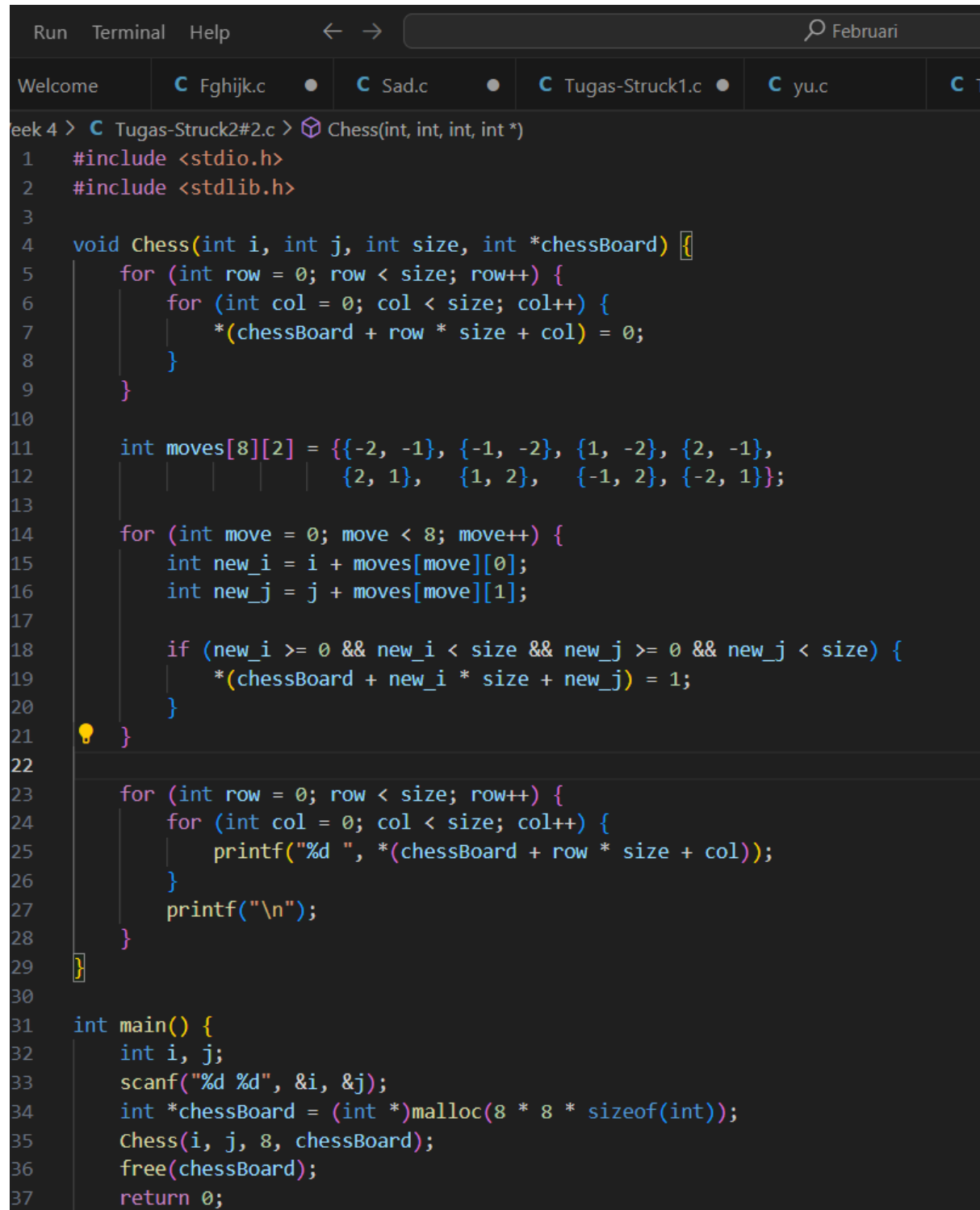


Nama:M.AGUNG BALYA

NIM:1203230095

Kelas:IF 03-01

Code



```
Run Terminal Help < -> Februari
Welcome C Fghijk.c C Sad.c C Tugas-Struck1.c C yu.c C T
Week 4 > C Tugas-Struck2#2.c > Chess(int, int, int, int *)
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void Chess(int i, int j, int size, int *chessBoard) {
5      for (int row = 0; row < size; row++) {
6          for (int col = 0; col < size; col++) {
7              *(chessBoard + row * size + col) = 0;
8          }
9      }
10
11     int moves[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1},
12                       {2, 1}, {1, 2}, {-1, 2}, {-2, 1}};
13
14     for (int move = 0; move < 8; move++) {
15         int new_i = i + moves[move][0];
16         int new_j = j + moves[move][1];
17
18         if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
19             *(chessBoard + new_i * size + new_j) = 1;
20         }
21     }
22
23     for (int row = 0; row < size; row++) {
24         for (int col = 0; col < size; col++) {
25             printf("%d ", *(chessBoard + row * size + col));
26         }
27         printf("\n");
28     }
29 }
30
31 int main() {
32     int i, j;
33     scanf("%d %d", &i, &j);
34     int *chessBoard = (int *)malloc(8 * 8 * sizeof(int));
35     Chess(i, j, 8, chessBoard);
36     free(chessBoard);
37     return 0;
```

Output

```

PS C:\agung\Februari> cd "c:\agung\Februa
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:\agung\Februari\Week 4>

```

Penjelasan

```

void Chess(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;
        }
    }
}

```

for (int row = 0; row < size; row++) untuk loop luar yang mengiterasi baris dari papan catur

for (int col = 0; col < size; col++) adalah loop dalam yang mengiterasi kolom dari papan catur untuk setiap baris.

*(chessBoard + row * size + col) = 0; untuk setiap iterasi nilai pada indeks yang sesuai di papan catur diatur menjadi 0

```

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

for (int move = 0; move < 8; move++) {
    int new_i = i + moves[move][0];
    int new_j = j + moves[move][1];

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

```

int moves[8][2] merupakan array dua dimensi untuk menampung semua kemungkinan Langkah yang dapat dilakukan oleh pion kuda.

for (int move = 0; move < 8; move++) adalah loop iterasi melalui semua kemungkinan Langkah yang mungkin dilakukan oleh pion kuda

int new_i = i + moves[move][0]; dan int new_j = j + moves[move][1]; berfungsi untuk menghitung koordinat baru (new_i, new_j) yang mewakili posisi setelah melakukan langkah sesuai dengan langkah yang sedang diperiksa.

if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) untuk memeriksa apakah posisi baru (new_i, new_j) ini memeriksa apakah posisi baru (new_i, new_j) berada dalam batas papan catur

*(chessBoard + new_i * size + new_j) = 1; jika posisi baru berada dalam batas papan catur, maka posisi tersebut ditandai dengan 1 di papan catur.

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

for (int row = 0; row < size; row++) adalah loop luar yang iterasi melalui semua baris pada papan catur.

for (int col = 0; col < size; col++) adalah loop dalam yang iterasi melalui semua kolom pada setiap baris papan catur.

```
int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int *chessBoard = (int *)malloc(8 * 8 * sizeof(int));
    Chess(i, j, 8, chessBoard);
    free(chessBoard);
    return 0;
}
```

int *chessBoard = (int *)malloc(8 * 8 * sizeof(int));: Menggunakan fungsi malloc untuk mengalokasikan memori dinamis yang cukup untuk menyimpan papan catur 8x8. Ukuran memori yang dialokasikan dihitung dengan mengalikan jumlah baris dan kolom dengan ukuran sizeof(int).

Chess(i, j, 8, chessBoard);: Memanggil fungsi kobaImaginaryChess dengan memberikan posisi bidak kuda (i dan j), ukuran papan catur (8), dan pointer ke papan catur (chessBoard) sebagai argumen.

free(chessBoard);: Menggunakan fungsi free untuk membebaskan memori yang dialokasikan untuk papan catur setelah selesai digunakan. Ini dilakukan untuk mencegah kebocoran memori.

return 0; menandakan bahwa program telah selesai dijalankan dengan sukses.