

No 1.

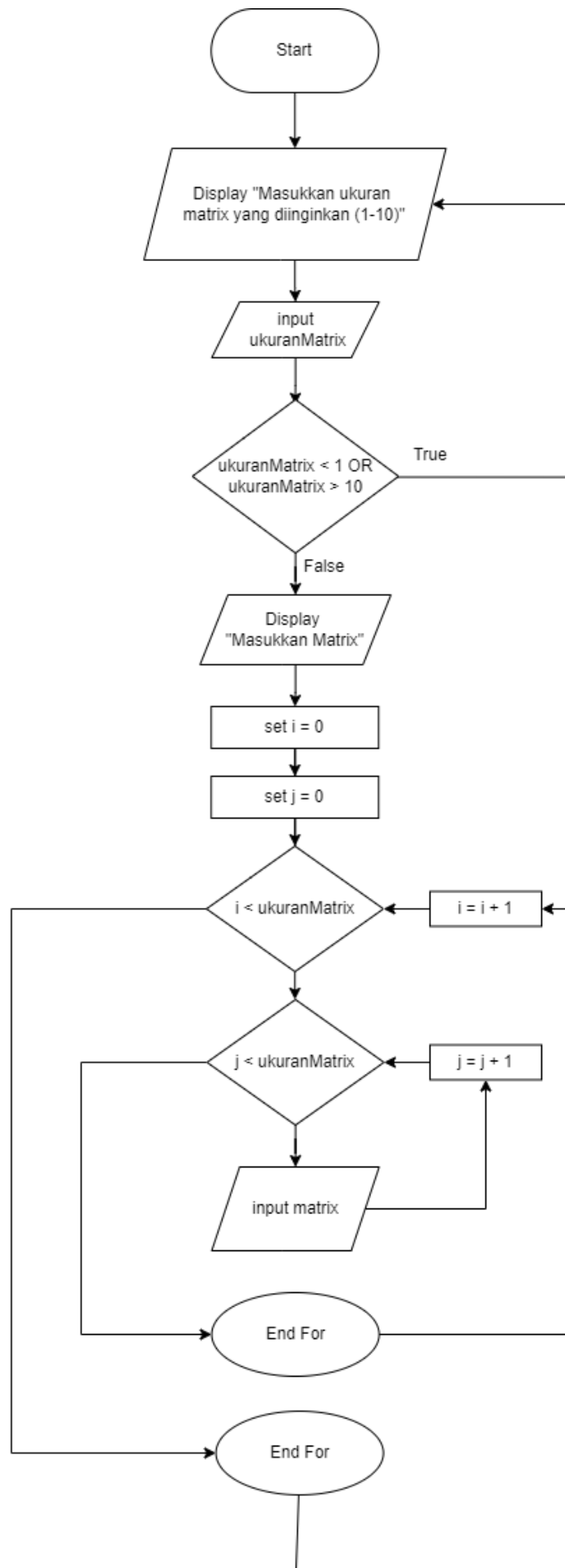
Narasi

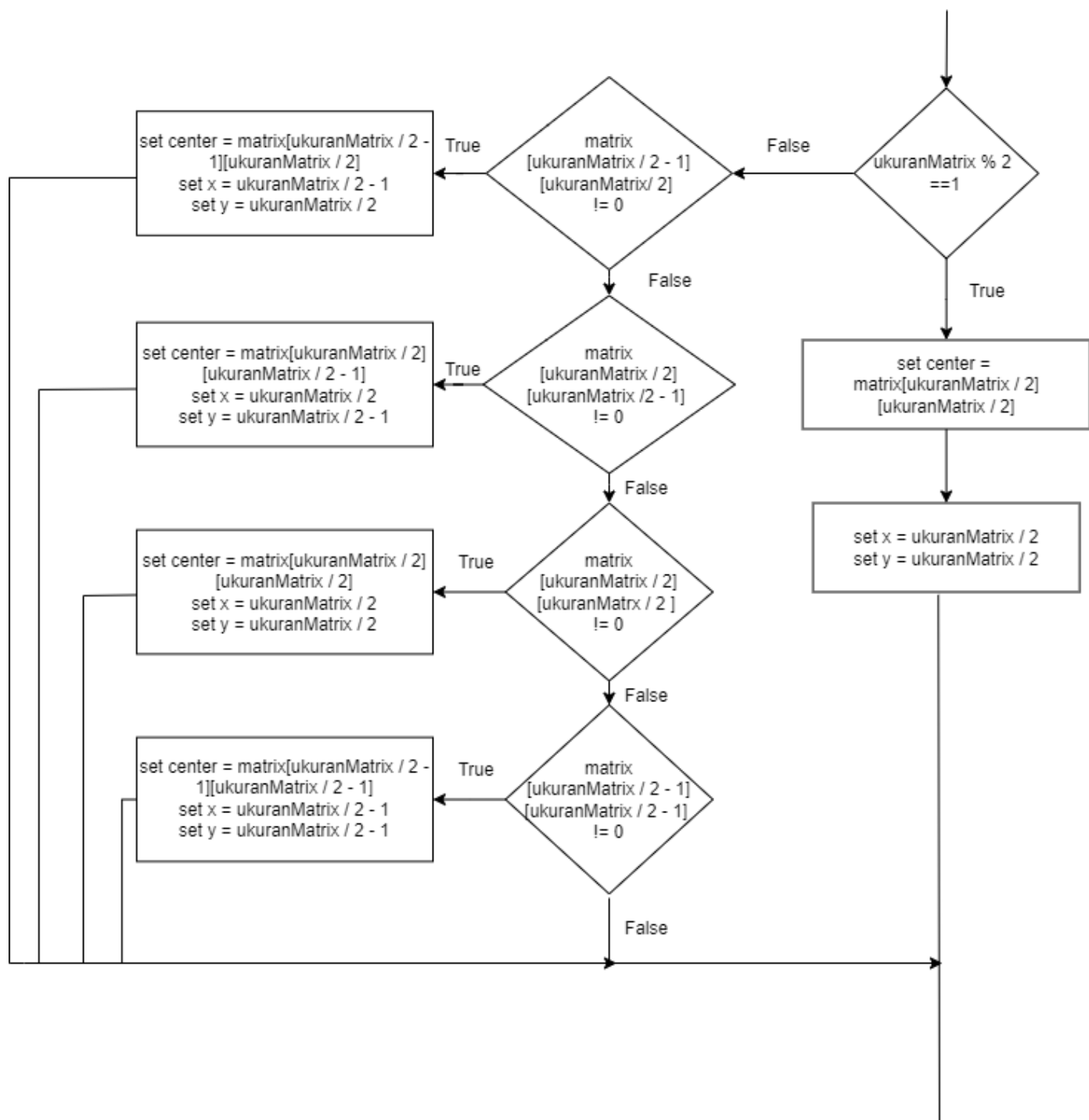
Untuk soal no.1, saya menggunakan library `stdio.h` dan `math.h`. Kemudian langkah pertama adalah scan variable `a` untuk ukuran matrix yang diinginkan. Disini saya memvalidasi inputan menjadi 1 sampai 10 karena saya membatasi ukuran matrix hanya mencapai 10x10. Setelah itu, membuat double for loop untuk scan isi matrix nya dengan variable `c` 2 dimensi (`c[a][a]`). Langkah selanjutnya, mendeklarasi variabel `center,x,y, o[100],p[100]` (untuk variabel `center` diinialisasi menjadi 0 dahulu).

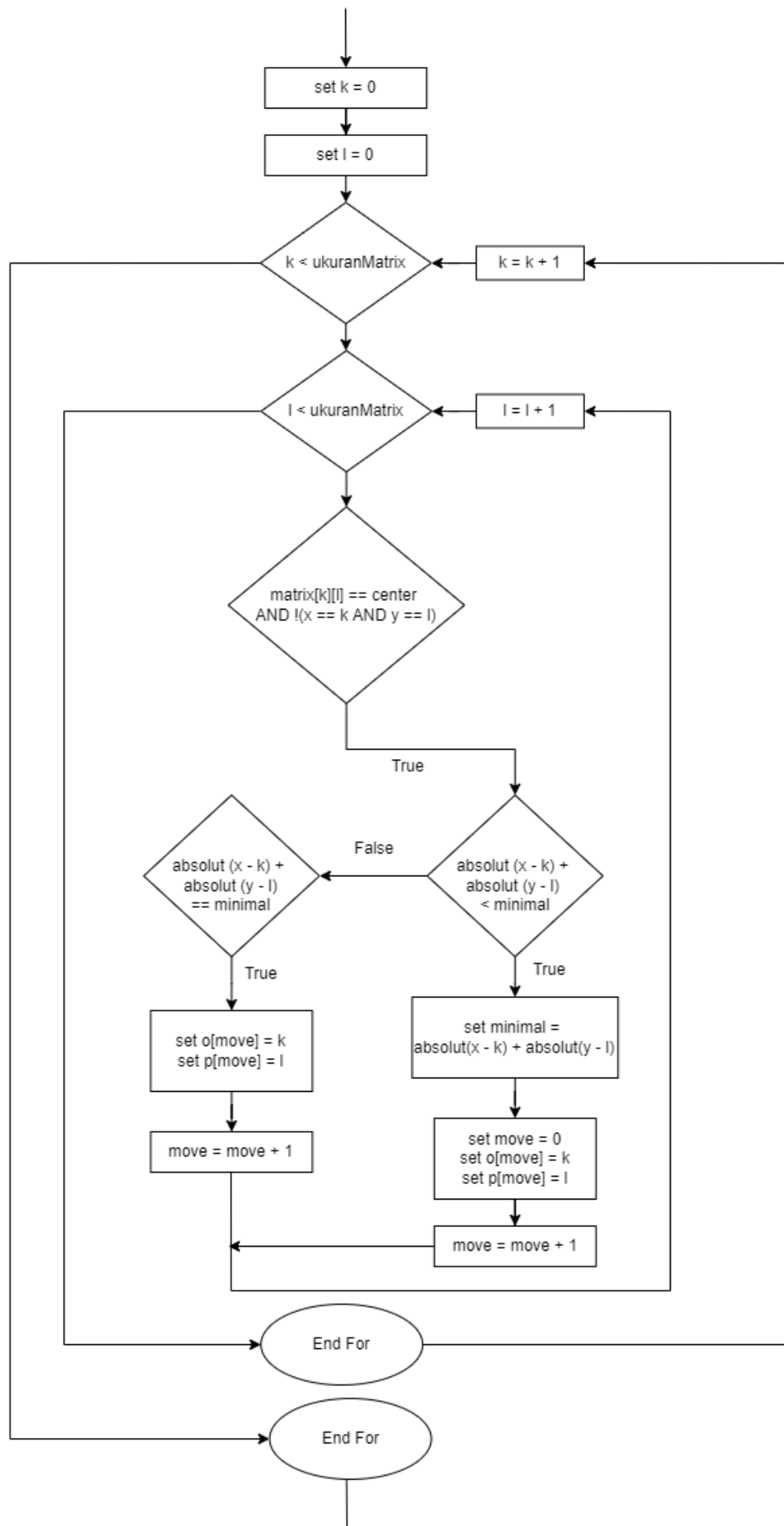
Selanjutnya masuk ke logika mencari `center`(titik tengah) dari suatu matrix. Disini menggunakan `if else` dimana jika ukuran matrixnya ganjil, maka titik tengahnya adalah `c[ukuran matrix / 2][ukuran matrix / 2]`. Misalnya ukuran matrix 5, maka `center` nya 2 (seharusnya 2.5 tetapi karena integer menjadi 2). Setelah menentukan `center`, simpan koordinat nya di variabel `x` dan `y` karena akan digunakan untuk mencari elemen terdekat. Sementara itu, jika ukuran matrixnya genap, maka titik tengahnya akan ada 4 situasi yaitu kanan atas, kiri bawah, kanan bawah dan kiri atas. Tiap situasi akan dicek apakah tidak sama dengan 0, jika benar maka akan disimpan koordinatnya di variabel `x` dan `y`. Untuk kanan atas adalah (`c[ukuran matrix / 2 - 1][ukuran matrix / 2] != 0`), kiri bawah adalah (`c[ukuran matrix / 2][ukuran matrix / 2 - 1] != 0`), kanan bawah adalah (`c[ukuran matrix / 2][ukuran matrix / 2] != 0`) dan kiri atas adalah (`c[ukuran matrix / 2 - 1][ukuran matrix / 2 - 1] != 0`).

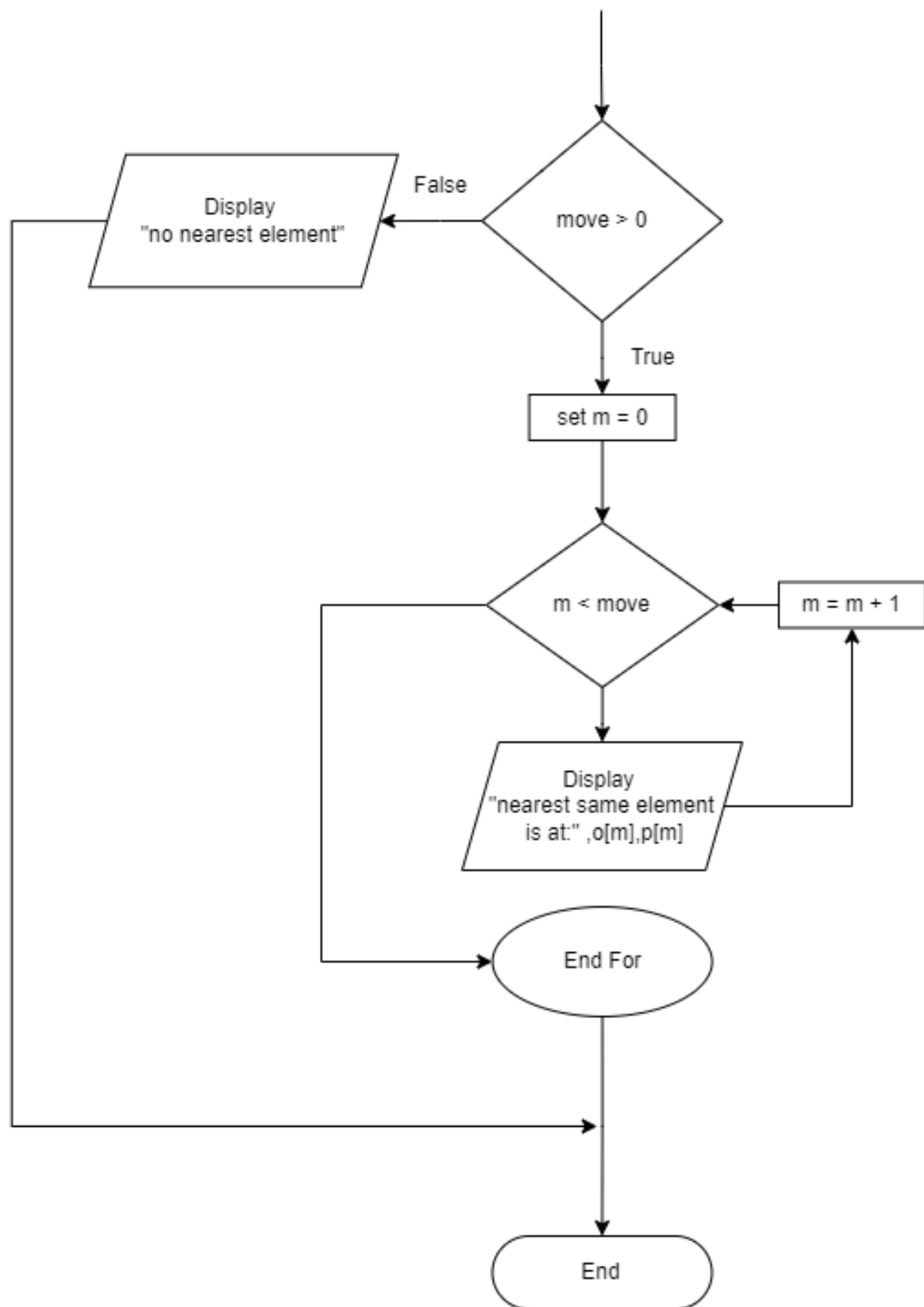
Selanjutnya masuk ke logika elemen terdekat. Pertama, mendeklarasi variable `move` dan `min` (`move` diinialisasikan menjadi 0 sedangkan `min` diinialisasikan menjadi 11). Kemudian, menggunakan double for loop untuk membaca matrixnya dan didalamnya terdapat `if(c[k][l] == center && !(k == x && y == l))`. Tujuannya adalah jika koordinat matrix yang sama dengan `center`(titik tengah) dan koor `x` dan `y` tidak sama dengan `k` dan `l`, maka terdapat 2 kondisi. Jika `absolut(x - k) + absolut(y - l)` kurang dari `min` (hal ini menyebabkan terdapat minimal `move` baru), maka variable `min = abs(x - k) + abs(y - l)` (diabsolut terlebih dahulu). Untuk mencari berapa `move` ke `center`, maka koor `x` dan `y` harus dikurang dengan `k` dan `l`. Kemudian variable `move` direset 0, variable `o[move]` dan `p[move]` menyimpan koordinat elemen terdekat yang sudah didapat dan terakhir variable `move` di increment. Sementara itu, jika `absolut(x - k) + abs(y - l)` sama dengan `min`, maka langsung variable `o[move]` dan `p[move]` menyimpan value `k` dan `l` yang sudah didapat dan variable `move` diincrement.

Langkah terakhir, jika `move` lebih dari 0, maka display koordinat elemen terdekat menggunakan loop `m` sebanyak `move`, sementara jika tidak lebih dari 0, display `no nearest element`.









No. 2

Narasi

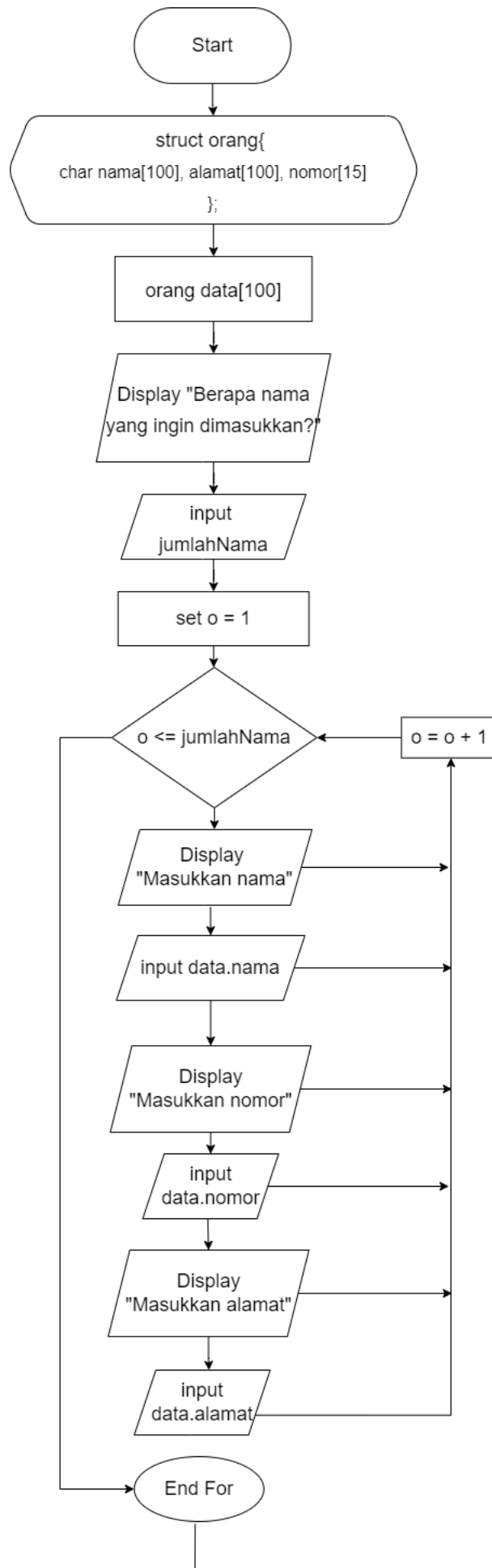
Untuk no.2 saya menggunakan library stdio.h dan string.h. Langkah pertama, membuat struct orang yang isinya ada char nama[100], char nomor[15] dan char alamat[100]. Selanjutnya, masuk ke int main dan mendeklarasi array orang data[100] untuk menampung nama, nomor dan alamat yang di input nantinya. Setelah itu scan variable n (berapa nama yang ingin dimasukkan). Lalu, terdapat for untuk melakukan loop sebanyak n dan isi for ada scanf nama, nomor dan alamat dari array datanya. Disini, scanf nama dan alamat menggunakan [`^\n`] supaya bisa membaca spasi di antara kata serta menggunakan `getchar()`. Setelah menginput data orang, dilanjutkan dengan scanf karakter dari nama yang ingin dicari dengan char search[10] dan ditambah `getchar()`.

Selanjutnya masuk ke logika search nama. Pertama, deklarasi integer `b = strlen(search)`. Tujuannya menghitung berapa panjang kata dari kata search yang di input sebelumnya. Kemudian deklarasi juga integer tanda3 dan di inialisasikan menjadi 0. Setelah itu, terdapat for loop i sebanyak n dan isinya terdapat deklarasi integer tanda = 0 dan terdapat integer a yang valuenya adalah panjang kata dari nama di array data yang sudah di input sebelumnya. Lalu, terdapat double for yaitu loop j sebanyak a (panjang kata nama di data) dan loop k dari 0 sampai `k < b && j + k < a`. Diantara loop j dan k, terdapat integer tanda2 yang di inialisasikan menjadi 0. Beralih ke loop k, disini looping akan dilakukan sebanyak b(panjang kata search) dan diberi syarat `j + k < a`, supaya tidak mengecek per huruf melewati batas nama yang di input. Misal "will", setelah l terakhir sudah stop di cek. Lalu, lanjut isi for terdapat satu if dimana jika nama dari array data sama dengan kata search yang kita mau cari, maka var tanda2 increment. Di perintah `if(data[i].nama[j + k] == search[k])`, index nama menggunakan + k karena digeser satu satu per huruf. Setelah dicek semua hurufnya dan increment tanda2 sudah selesai, dilanjutkan dengan 1 kondisi diluar loop k dimana jika tanda2 sama dengan b(panjang kata search), maka tanda sama dengan 1 dan di break.

Selanjutnya keluar dari loop j, terdapat 1 kondisi dimana jika tanda sama dengan 1, maka display nama, nomor, alamat dari array data yang diminta. Setelah display, var tanda3 diinialisasikan menjadi 1. Keluar dari loop i, terdapat kondisi dimana jika tanda sama dengan 1, maka display data not exist.

Flowchart

No.2



No. 3

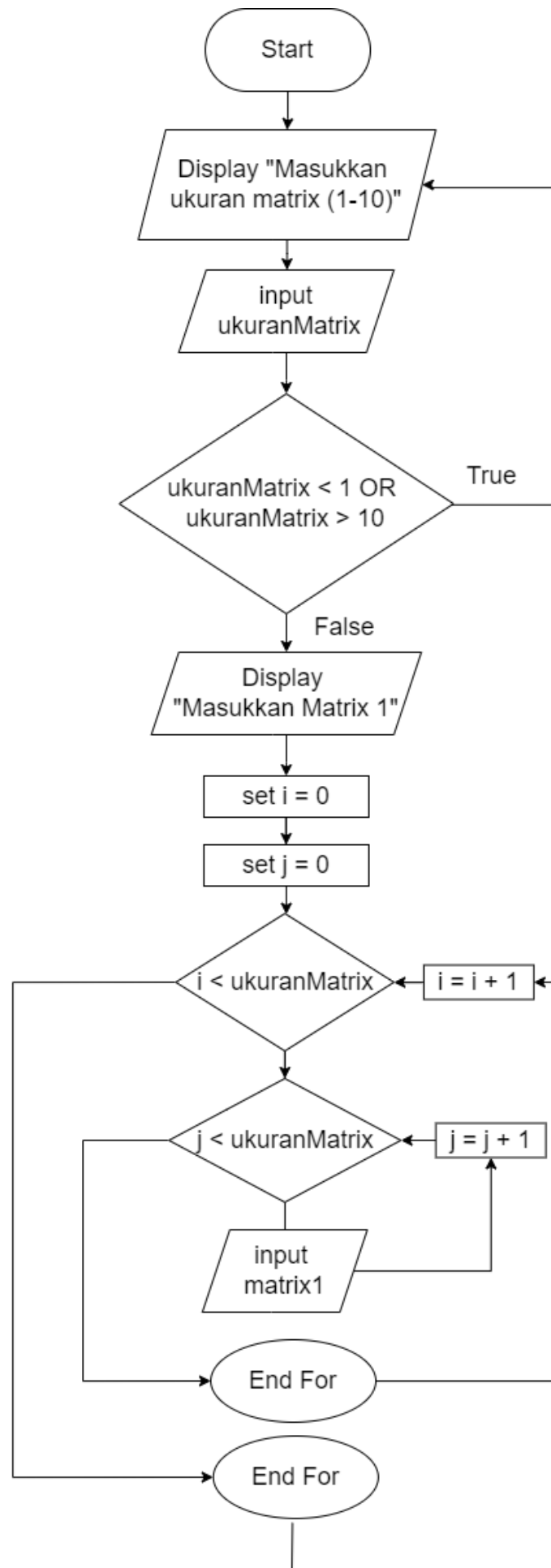
Narasi

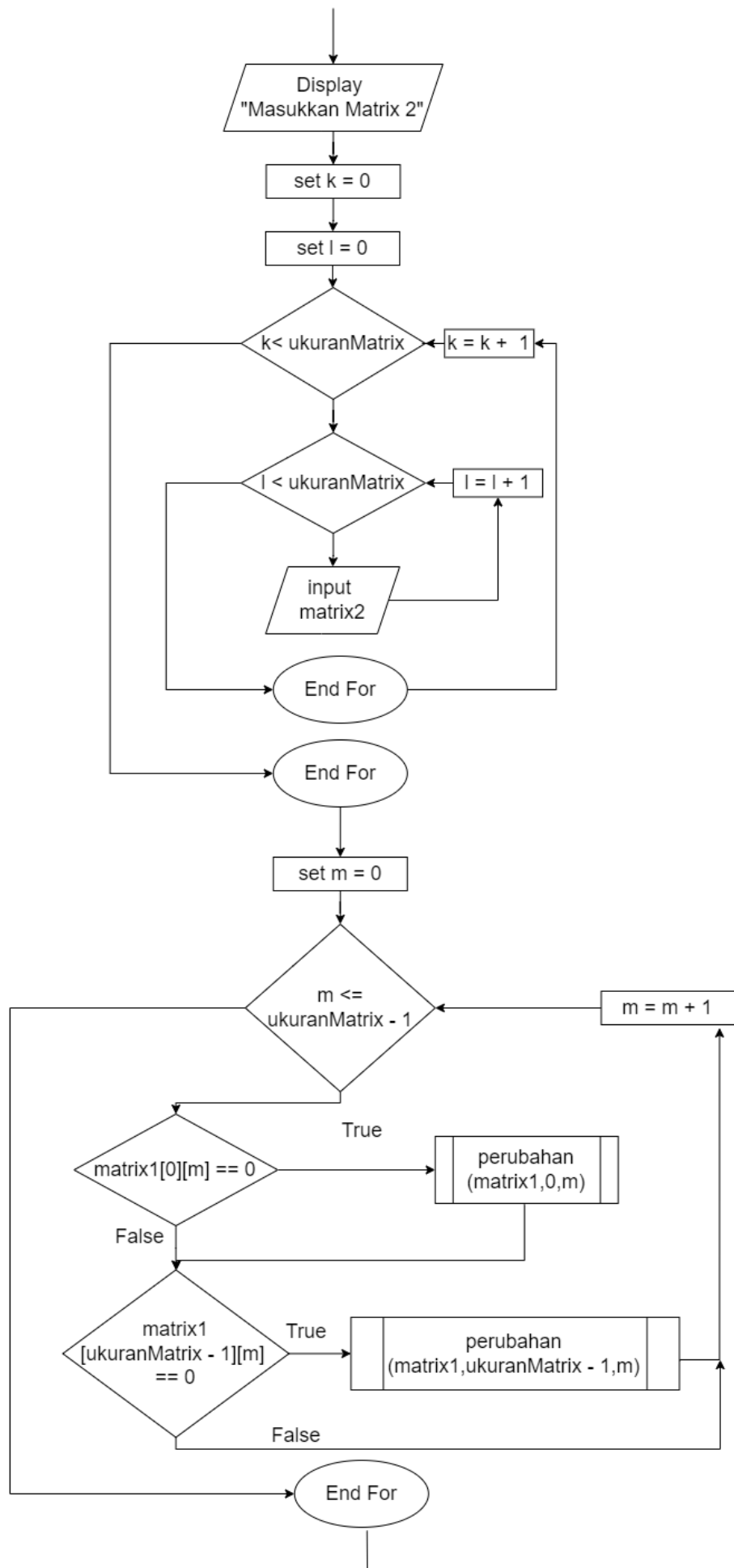
Untuk no.3 saya menggunakan library studio.h. Langkah pertama, mendeklarasi variable a menjadi global variable. Kemudian, memvalidasi ukuran matrix yang di input user. Disini saya memberi batasan dari 1 – 10 untuk ukuran matrix. Lalu, mendeklarasi 2 dimensi array b dan c serta melakukan input untuk matrix 1 dan 2 di dalam double for loop sesuai ukuran yang sudah di input sebelumnya. Selanjutnya, terdapat logika untuk membaca matrix 1 dan matrix 2 yang meliputi baris atas, baris bawah, kolom kiri dan kolom kanan. Untuk membaca baris atas dan bawah pada matrix 1 dan matrix 2, kita melakukan looping dari 0 sampai dengan a – 1 (ukuran matrix kurang 1). For pertama (loop m dan o) menggunakan tanda sama dengan supaya angka dipojok tidak di baca. Setelah for ditulis, terdapat 2 situasi, jika di baris atas terdapat angka 0, maka akan memanggil function perubahan. Begitu juga dengan situasi 2, jika baris bawah terdapat angka 0, maka akan memanggil function perubahan. Untuk for kedua (loop n dan p), kita melakukan looping dari 1 sampai a – 1 (ukuran matrix kurang 1). Bedanya disini tidak ada tanda sama dengan karena akan membaca angka di pojok matrix. Setelah for akan ada 2 situasi dimana situasi pertama, jika kolom kiri terdapat angka 0, maka akan memanggil function perubahan. Begitu juga dengan kolom kanan, jika ada angka 0, maka akan memanggil function perubahan.

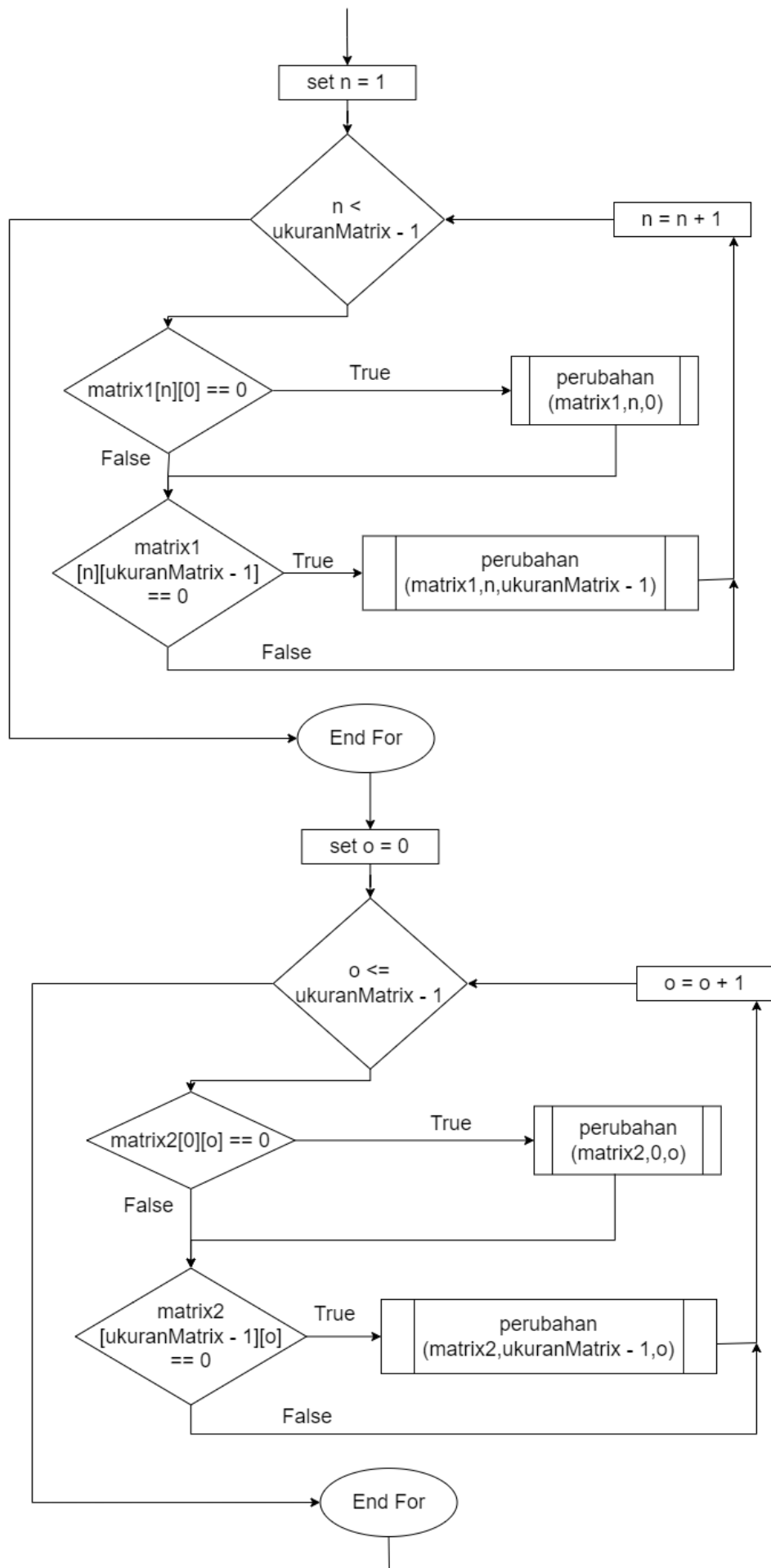
Selanjutnya adalah function perubahan. Difunction ini, saya menggunakan parameter b[11][11], x dan y. Logika dari function ini menggunakan rekursif dan terdapat 2 kondisi, yang pertama adalah jika $x < 0$ OR $y < 0$ OR $x \geq a$ OR $y \geq a$ atau dengan kata lain adalah diluar dari batas matrix yang ditentukan, maka akan return karena value yang diberi salah. Sementara, kondisi kedua adalah jika value nya masuk dalam matrix, maka langkah pertama yang dilakukan adalah convert angka 0 yang ada menjadi angka 1 karena yang di inginkan adalah 0 di dalam angka 1. Kemudian terdapat 4 kondisi, disini akan mengecek apakah di atas, kiri, bawah dan kanan terdapat 0 atau tidak. Jika iya, maka akan memanggil function perubahan dan begitu seterusnya.

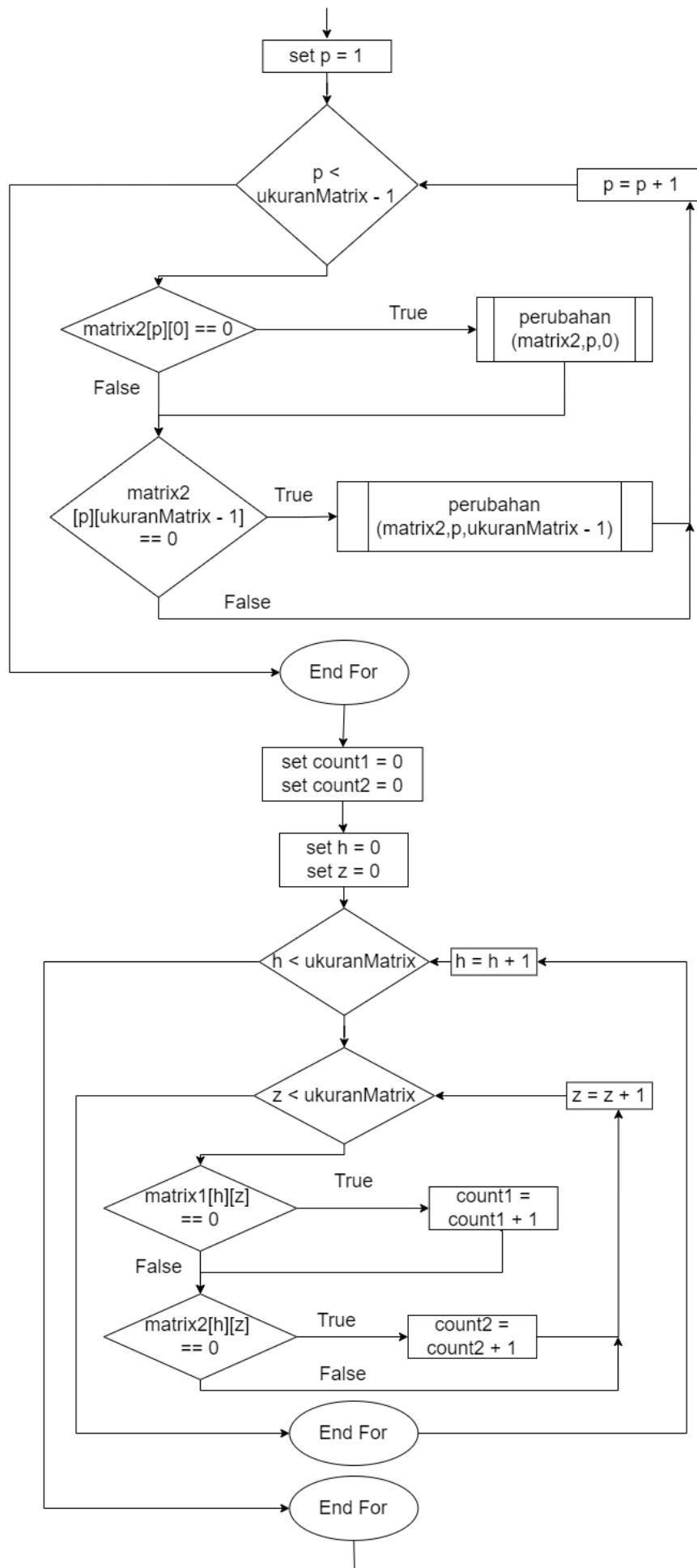
Selanjutnya adalah mencari counternya. Langkah pertama mendeklarasikan count1 dan count2 lalu diinialisasikan menjadi 0. Lalu terdapat double for loop untuk membaca keseluruhan matrix. Kemudian terdapat 2 kondisi, jika di matrix b ada yang memiliki angka 0, maka count1 increment. Sementara kondisi 2, jika matrix c ada yang memiliki angka 0, maka count2 increment. Setelah sudah mengetahui berapa angka 0 disetiap matrix, dilakukan if else. Jika count1 lebih besar dari count 2, maka display object 1 is bigger. Jika count1 lebih kecil dari count 2, maka display object 2 is bigger. Jika count1 dan count2 sama, maka display object 1 and 2 has the same size.

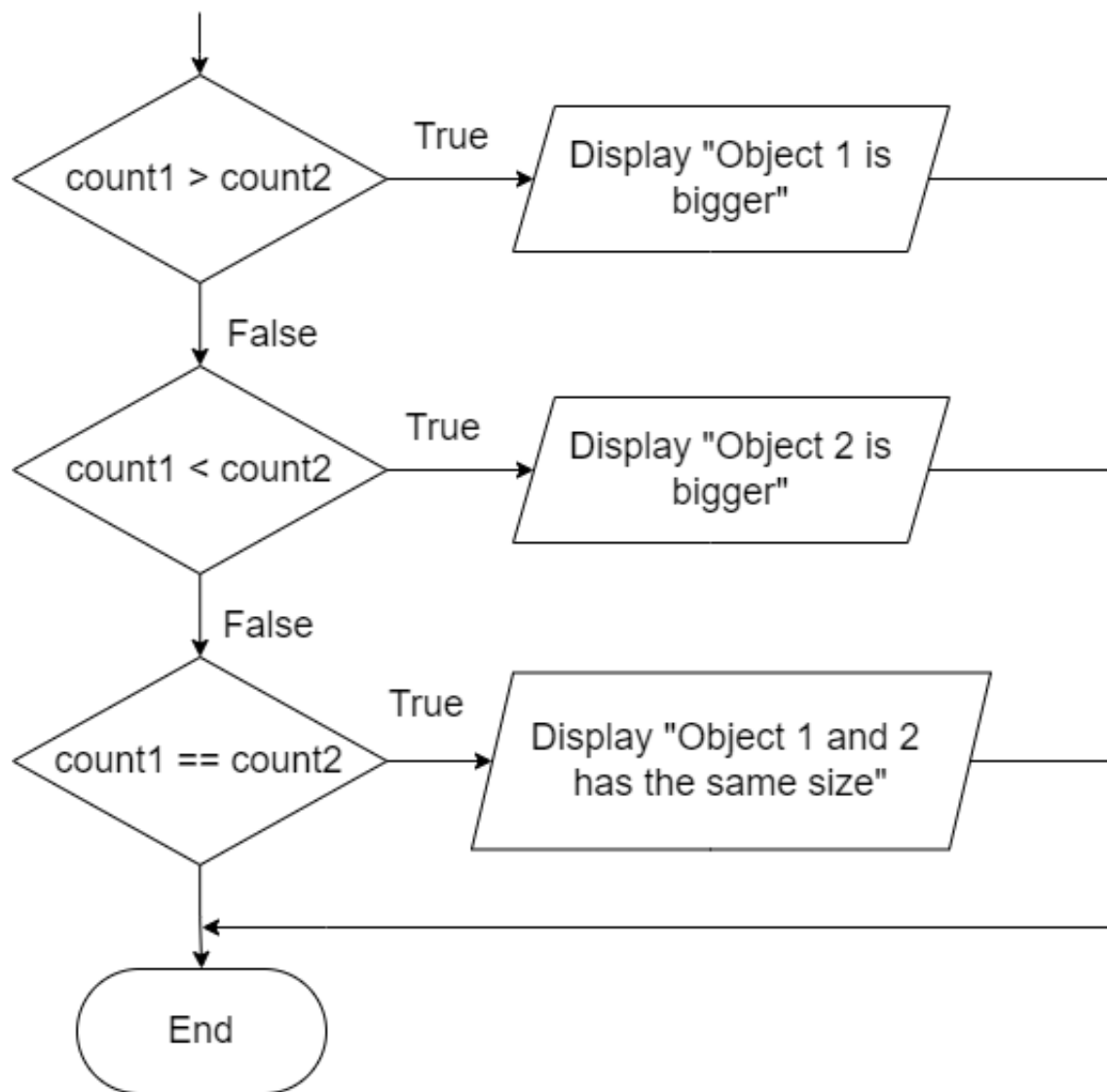
No 3











Function

