

**BLM3021**

**ALGORİTMA ANALİZİ**

**ÖDEV-1**

**SORU 2**

**17011701 Onur KOÇKAN**

**Problem 2:**Bir matriste, verilen N değeri dikkate alınarak, “Von Neumann’s Neighborhood” kuralına göre(https://mathworld.wolfram.com/vonNeumannNeighborhood.html)hücrelerin “1” değeri ile işaretlenmesi isteniyor. Aşağıda N=0,1,2 değerlerine göre matris içerisindeki hücreler siyahile gösterilmektedir (Beyaz=”0”, Siyah = “1”).

Von Neumann’s Neighborhood” kuralını gözeterek verilen N değerine göre matrisleri oluşturan programın kodunu C dilinde yazınız. Program çıktısı,ilgili matrisi ve satır başına ve toplam siyah hücre sayısını göstermelidir. Matris dinamik bellek ayırma yöntemi ile en uçtaki siyah hücrelerin dışında sadece bir satır ve sütun kalacak şekilde oluşturulmalıdır. Örnek N=0 için 3x3, N=1 için 5x5 boyutundaolmalıdır.Şekildeki matris boyutlarını DİKKATE ALMAYINIZ.**(60 Puan)**

****

**ÇÖZÜM**

Kodlar:

#include <stdio.h>

#include <stdlib.h>

int bCount**[**250**]={**0**};**//bu dizi ile elde edilen siyah adetlerin saym yaplr.Initiate de𥲠olarak 0 verilmitir.

void createArrayWithZeros**(**int **\*\***array**,**int arrayLenght**){**

int i**,**j**;**

**for(**i**=**0**;**i**<**arrayLenght**;**i**++){**

**for(**j**=**0**;**j**<**arrayLenght**;**j**++){**//matrisimizin tm elemanlarn 0lar ile dolduruyoruz.

array**[**i**][**j**]=**0**;**

**}**

**}**

**}**

void fillArrayWithOnes**(**int **\*\***array**,**int arrayLenght**,**int N**){**

int i**,**j**;**

**for(**i**=**0**;**i**<**arrayLenght**;**i**++){**

**for(**j**=**0**;**j**<**arrayLenght**;**j**++){**

**if(**abs**(**i**-**N**-**1**)+**abs**(**j**-**N**-**1**)<=**N**){**//gerekli kontrol yaplyor.ޡrt sa𬡮yor ise;

array**[**i**][**j**]=**1**;**//ilgili hcrenin de𥲩 1leniyor.

bCount**[**i**]++;**//o satira ait sayacn de𥲩 1 arttrlyor.

**}**

**}**

**}**

**}**

void showArray**(**int **\*\***array**,**int arrayLenght**,**int N**){**

int i**,**j**;**

**for(**i**=**0**;**i**<**arrayLenght**;**i**++){**//basit ekilde arrayimizi print etti𩭩z kod blo𵮍

**for(**j**=**0**;**j**<**arrayLenght**;**j**++){**

printf**(**"%d "**,**array**[**i**][**j**]);**

**}**

printf**(**"\n"**);**

**}**

printf**(**"\n"**);**

**for(**i**=**0**;**i**<**arrayLenght**;**i**++){**//satr says kadar,satrlarda ka砳iyah(1) adeti bilgisini ekrana bastryoruz.

printf**(**"Satir NO=%d <-> Siyah kare adeti=%d \n"**,**i**,**bCount**[**i**]);**

**}**

printf**(**"Toplam siyah kare adeti=%d"**,(**2**\***N**\*(**N**+**1**))+**1**);**//toplam siyah kare adetinide dantasyonunda refere edilen kaynaktaki formle goluturup ekrana bastryoruz.

**}**

int main**(**int argc**,** char **\***argv**[])** **{**

int N**,**arrayLenght**,\*\***array**;**//gerekli degisken tanmlamalarmz.

printf**(**"N degerini girin:"**);**//N degeri kullancdan istenilir ve degiskene atilir.

scanf**(**"%d"**,&**N**);**

arrayLenght**=(**2**\***N**)+**3**;**//verilen N degiskenine gdizinin boyutunu belirleyecek formul ile dizinin boyutu belirlenir.

// danndaki 𩠢az alrsak.N=0 i穮 3x3,N=1 i穮 5x5 vs... sa𬽹ordur.

array**=(**int**\*\*)**calloc**(**arrayLenght**,sizeof(**int**\*));**//dizi i穮 bellekte alan istenmesi.(4\*diziBoyutu) Byte

int i**,**j**;**

**for(**i**=**0**;**i**<**arrayLenght**;**i**++){**

array**[**i**]=(**int**\*)**calloc**(**arrayLenght**,sizeof(**int**));**

**}**

createArrayWithZeros**(**array**,**arrayLenght**);**//0lar ile doldurarak matrisi oluturacagmz fonksiyon.

fillArrayWithOnes**(**array**,**arrayLenght**,**N**);**//gerekli yerleri tespit edip 1 de𥲬erini verecek fonksiyon.

showArray**(**array**,**arrayLenght**,**N**);**//arrayimizi gricek fonksiyon.

free**(**array**);**//btn ilemler sonunda arraymz bellekten freeliyoruz.

**return** 0**;**

**}**

Ekran çıktıları:





