

BİLGİSAYAR PROGRAMLAMA III DERSİ

PROJE 1 RAPORU

Projede Yaptıklarım :

Projeme ilk olarak “matrisoperations.c” dosyayı oluşturarak başladım. Projemde matrisleri random olarak fonksiyonlar ile oluşturmayı tercih ettim. Bu yüzden ilk olarak ‘makematrix’ fonksiyonumu oluşturdum. Projede matriksler kare matriks olmak zorunda idi. Bu yüzden kullanıcıdan 2 kere aynı sayıyı almak yerine tek 1 sayı almayı ve işlemlerimi ona göre oluşturmayı düşündüm. Bu yüzden makematrix metoduma sadece 1 adet boyut belirten parametre yolladım. Bu metod aldığı parametreye göre malloc ile heap’ ten yer ayırıyor ve heaptan matrisimiz için ayırdığı yerin adresini geri döndürüyor. Daha sonra heaptan yer ayırarak oluşturduğum matrislerimin içini random olarak doldurmak için fillmatrix metodumu yaptım. Bu metod parametre olarak dolduracağı matrisin adresini (pointer ile) ve o matrisin boyutunu alıyor. Çıktılarım doğru mu diye kolaylıkla hesaplayabilmek için random sayı aralığımı 1-10 arasında tuttum. Bu metod sadece heapteki ilgili yerdeki elemanları değiştiriyor, yani herhangi bir geri dönüş yapmıyor.

Matrislerimi oluşturup doldurduktan sonra kullanıcının oluşan matrisleri görebilmesi için printmatrix metodumu yaptım. Bu metod parametre olarak ekrana basılacak olan ilgili matrisin adresini (double pointer) ve boyutunu alıyor. Aldığı boyuta göre ilgili matrisin elemanlarını tek tek alıp ‘a’ değişkenine atadım ve ‘a’ değişkeninin değerini bastırdım. İlgili sütunun sonuna gelince bir alt satıra geçmesi için de bir ‘if’ kontrolü yaptım.

Bu aşamadan sonra projemin asıl kısmı olan operasyon metodu için ihtiyacım olan fonksiyonları tasarlamaya başladım. İlk önce dolgu değeri için proje taslağında verilen bir formül vardı. Kısa olduğu için ilk olarak ‘valueofd’ fonksiyonunu tasarladım. Bu fonksiyon içerisine küçük olan matrisin boyutunu alıyor ve buna göre ilgili formül ile dolgu değerini hesaplayıp geri döndürüyor.

Daha sonra matris 3 için boyut belirleyen ‘calculatematris3size’ fonksiyonunu yazdım. Bu fonksiyon içine matris 1’ in ve matris 2’ nin boyutunu ve başka bir fonksiyonda hesaplanmış olan dolgu değerini alıyor. Daha sonra fonksiyonum bu parametrelerin değerleri ile ilgili formülü uygulayıp matris 3’ ün boyut değerini geri döndürüyor.

Bütün bunlardan sonra asıl metodum olan ‘operations’ metodumu oluşturdum. Bu metodu ilk olarak birçok parametre ile kağıtta deneyerek algoritmasını oluşturdum ve daha sonra koda dökmeye başladım. Bu metod parametre olarak matris1 ve matris 2 nin adreslerini (double pointer) alıyor. Bunların dışında fonksiyonum bu matrislerin boyutlarını ve kullanıcının tercih ettiği işlem tipini de parametre olarak alıyor. Bu aşamada işlem tipi çok önemli. Bu yüzden burada bir kontrol yaptım. İşlem tipi ‘1’ ise ,dolgu değeri fonksiyonum yardımıyla, dolgu değeri tekrar hesaplanmalı. Ayrıca işlem tipi 1 olduğunda matris 1’in boyutu dolgu değerine göre değişmeli (bu fonksiyonu daha sonra yaptım). Fakat işlem tipim ‘0’ ise dolgu değerim otomatik olarak 0 olmalı. If Else yapısı ile işlem tipine göre dolgu değerimi ve matris 1’i güncelledim. Daha sonra girilen matris1 ve matris2 boyutlarını ‘calculatematris3size’ fonksiyonuma yollayarak matris 3’ ün boyutunu hesapladım ve bu boyut yardımıyla makematrix fonksiyonumla malloc ile heaptan bu matrisime yer ayırdım. Daha sonra küçük fonksiyonu büyük fonksiyon üzerinde gezdirip çarpım ve toplam işlemlerini yapabilmek için dörtlü bir for sistemi kullandım. Bu sistemde dıştaki for döngüleri (i ve j değişkenli olanlar) küçük matrisin büyük matris üzerinde gezmesini sağlıyorlar. Burada i ve j değişkenleri için sınır noktam matris 3 ün boyutu. Çünkü yaptığım hesaplama ve denemelere göre küçük matris büyük matris üzerinde en fazla matris 3 ün sütun sayısının 1 eksiği kadar ileri gidebiliyor. Bu şart satır durumu için de geçerli. İç kısma koyduğum diğer iki for döngüsü (k ve q) küçük matris büyük matrisin üzerinde ilgili yere yerleştikten sonra elde edilen alanda elemanları tek tek (indeks indeks) (Burada küçük matrisin indeksleri k ve g değerlerinden i ve j kadar eksik oluyor, bunun nedeni ileri giden indeks sayısını geri çekmektir.) çarpıp sonuçları toplama görevini görüyor. Toplanan sonuçları da sırasıyla matris 3 ün i. ve j. elemanlarına atadım. Son olarak da elde ettiğim matrisi ekrana bastırdım ve matris 3’ ün adresini fonksiyondan geri döndürdüm.

İşlem tipinin 1 olması durumunda matris 1' in boyutu dolgu değerine göre değişmeliydi. Bunun için yeni bir 'newMatriswD' fonksiyonu yaptım. Bu fonksiyon parametre olarak değişecek olan matrisi, bu matrisin boyutunu ve dolgu değerini alıyor. Dolgu değeri 1 olduğunda matrisin çevresine bir sıra, 2 olduğunda 2 sıra sıfır gelmeli ve örüntü diğer dolgu değerleri için de aynı sırada devam etmeli. Programımda hep malloc kullandığım için farklı türleri de kullanabilmek adına bu kısmı calloc ile yaptım ki sıfırlar otomatik olarak çevreye dizilsinler. Bunun için öncelikle yeni boyut hesapladım. Hem alta hem de üste birer sıra ekleneceği için eski boyut artı iki çarpı dolgu değeri şeklinde yeni boyut hesapladım ve bu boyuta göre calloc ile geçici bir matris oluşturdum. Dolgu değerinden gelen sıfır sıraları haricindeki yerlerde matris 1' in elemanları bulunmalı. Bunun için iç içe for döngüsü kullandım. En sol sütunlar ve en üst satırlar için dolgu değerine kadar olan satır ve sütunlar 0 ile doldurulmalı. Aynı şekilde en sağ sütunlar ve alt satırlar için de yeni boyut – dolgu değerinden gelenler 0 ile doldurulmalı. Bunların dışında kalan yerlere ise matris 1' in elemanları atanmalı. Burada önemli olan kısım yeni matrisin elemanlarının indekslerinin (i ve j) matris 1' in elemanları ile aynı olmasıdır. Bu yüzden matris1 den atama yaparken indekslerden dolgu değerlerini çıkararak atama yaptım ve oluşan temp matrisini fonksiyondan geri döndürdüm.

Aslında bu aşamalardan sonra projemin işlemleri bitmişti fakat ben matris boylarını ve işlem tipini kullanıcı girişi yaptım. Bunun için ilk olarak 'matrixsize' metodu yaptım. Matris boyutları çift olamaz bu fonksiyon onu kontrol ediyor ve buna göre bir değer döndürüyor. Ayrıca matris 1' in boyutu 3'ten büyük ve tek olmalı. Bunun için de 'size1_control' metodu yaptım ve kullanıcıya bu konuda bilgilendirme yaptım. 'Size2_control' metodunda da matris 2' nin boyutunun 1 den büyük bir tek sayı olmasını ve aynı zamanda matris 1' in boyutundan küçük olmasını kontrol ettim. Daha sonra da işlem tipi sadece 0 ya da 1 olabilirdi. Bunun kontrolünü yapmak için 'processtype_control' fonksiyonumu hazırladım.

Son fonksiyon olarak da free işlemlerimi yapmak için letfreematrix metodumu yaptım. Bu metodda da heapten aldığım yerleri index index iade ettim.

Daha sonra bu '.c' dosyama kaynaklık edecek "matrisoperations.h" kütüphanemi tanımladım ve kütüphanenin içerisine fonksiyon imzalarını yazdım. Son olarak da bütün bu metodlarımı ilgili sırada "main.c" dosyamda çağırarak projemi tamamladım.

Ödevde dikkat ettiğim kadarıyla eksik bıraktığım bir yer bulunmamaktadır.

Ödevde zorlandığım kısımlar :

Ödevde ilk başta 4' lü for algoritmasını mantık açısından kurmakta zorlandım. Daha sonra kağıt üzerinde çokça deneme yaparak bu yapıyı çözdüm.

Program Çıktıları :

```
C:\bp3_proje>MinGW32-make
gcc main.c matrisoperations.h matrisoperations.c -o Proje1

C:\bp3_proje>Proje1
Argc has to be 5.
C:\bp3_proje>Proje1 5 11 0 200
Please enter the size of the 2nd matrix (Size1 has to bigger than size 2 and Size2 has to be greater than 1!!)
1
Please enter the size of the 2nd matrix (Size1 has to bigger than size 2 and Size2 has to be greater than 1!!)
3
CALCULATING BEGIN
First matrix :
 2      1      1      2      7
2      4      7      1      5
1      3      8      7      3
2     10      7      3      3
8      9      6      3      1

Second matrix :
 2      1      1
2      7      2
4      7      1

Matris 3 as a result :
85     139    126
139    184    143
202    177    113

C:\bp3_proje>
```

```

C:\bp3_proje>Proje1 7 5 1 200
CALCULATING BEGIN
First matrix :
 2   1   1   2   7   2   4
7   1   5   1   3   8   7
3   2  10   7   3   3   8
9   6   3   1   4   1   8
1   2   2   7   8   3  10
9   8   6   6   5   8   4
5   6  10   5   7  10   8

Second matrix :
 2   1   1   2   7
2   4   7   1   5
1   3   8   7   3
2  10   7   3   3
8   9   6   3   1

Matrix 1 after operations :
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0
0   0   2   1   1   2   7   2   4   0
0   0   7   1   5   1   3   8   7   0
0   0   3   2  10   7   3   3   8   0
0   0   9   6   3   1   4   1   8   0
0   0   1   2   2   7   8   3  10   0
0   0   9   8   6   6   5   8   4   0
0   0   5   6  10   5   7  10   8   0
0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0

Matris 3 as a result :
127 198 245 327 353 301 279
230 315 395 371 402 336 314
258 358 456 390 438 443 408
343 399 503 518 589 457 418
352 512 582 573 657 511 494
290 371 442 446 515 387 346
232 332 358 332 369 299 200

```

```

C:\bp3_proje>Proje1 5 3 9 200
!!Process type just could be 0 or 1 , please enter again !! : 8
!!Process type just could be 0 or 1 , please enter again !! : 1
CALCULATING BEGIN
First matrix :
 2   1   1   2   7
2   4   7   1   5
1   3   8   7   3
2  10   7   3   3
8   9   6   3   1

Second matrix :
 2   1   1
2   7   2
4   7   1

Matrix 1 after operations :
0   0   0   0   0   0   0
0   2   1   1   2   7   0
0   2   4   7   1   5   0
0   1   3   8   7   3   0
0   2  10   7   3   3   0
0   8   9   6   3   1   0
0   0   0   0   0   0   0

Matris 3 as a result :
34   56   79   70   92
35   85  139  126  97
43  139  184  143  75
103 202  177  113  63
86  112  96   55  22

```

Not : Yazdığım fonksiyonlar daha büyük dolgu değerleri için de çalışabilmektedir.

Hazırlayan : Feyza KURUÇAY

1921221009

Bilgisayar Mühendisliği / 2. Sınıf