**ALGORİTHM ÖĞRENCİ - DERS TAKİP SİSTEMİ**

struct ogrenci

{

ogr\_no , ad\_soyad[30],

sinif , kredi,not\_ort

struct ogrenci \*sonraki

}

struct notlar

{

ogr\_no , notu

struct notlar \*noya\_gore\_sonraki

struct notlar \*nota\_gore\_sonraki

struct notlar \*nota\_gore\_onceki

}

struct ders

{

ders\_kodu , ders\_adi[25] ,

kredi , ogr\_sayisi , not\_ort

struct notlar \*not\_bilgisi

}

int menu\_secim()

int ders\_kodu\_al()

int ogr\_no\_al()

int hashing(int no)

struct ogrenci \*ara(struct ogrenci \*, int )

void ogrenci\_sirali\_ekle(struct ogrenci \*\*, struct ogrenci \*)

int ders\_kodu\_belirle(struct ders \*[],int )

void ogr\_nota\_gore\_ekle(struct notlar \*,struct notlar \*)

void ogr\_noya\_gore\_ekle(struct notlar \*,struct notlar \*)

struct ders \*koda\_gore\_ara(struct ders \*[], int ,int )

void tasi(struct notlar \*,struct notlar \*)

void ders\_bilgileri\_goruntule(struct ders \*[])

struct ogrenci \* sinifa\_gore\_listele(struct ogrenci \*[], int ,int )

void notlar\_listesinden\_noya\_gore\_cikar(struct notlar \*,int )

void notlar\_listesinden\_nota\_gore\_cikar(struct notlar \*,int)

struct notlar \*ders\_icinde\_ogr\_arama(struct notlar \*,int)

struct notlar \*ders\_kodsuz\_ogr\_no\_ara(struct notlar \*, int )

void ogrenci\_listesinden\_cikar(struct ogrenci \*\*,int )

void not\_guncelle(struct ogrenci \*,struct ders \*,int ,int )

**START**

ogr\_say🡨0,altmis\_say🡨0,sayaç🡨0,sayac\_sinir\_alti🡨0,sayac\_1🡨0

not\_ortalama,toplam🡨0

struct ogrenci \*ogrenciler[100], \*bir\_ogrenci,\*gecici,\*bir\_ogr

struct ders \*dersler[89], \*bir\_ders

struct notlar \*bir\_not,\*liste\_dugumu,\*gecici\_ogr,\*bir\_no

Repeat for i 🡪1 to 100

ogrenciler[i]🡨NULL

end repeat

Repeat for i🡪1 to 89

dersler[i]🡨NULL

end repeat

Repeat do

Seçenek🡨menu\_secim()

Select seçenek from

case ‘1’:

bir\_ogrenci🡨malloc(sizeof(struct ogrenci)

if (bir\_ogrenci)

then a🡨ogr\_no\_al()

kod🡨hashing(a)

if (ara(ogrenciler[kod],a)=NULL)

then bir\_ogrenci ->ogr\_no🡨a

write "ogrencinin adini ve soyadini giriniz:"

fflush(stdin)

gets(bir\_ogrenci->ad\_soyad)

write "ogrencinin sinifi:"

read bir\_ogrenci->sinif

bir\_ogrenci->kredi🡨0

bir\_ogrenci->not\_ort🡨0.00

ogrenci\_sirali\_ekle(&ogrenciler[kod],bir\_ogrenci)

else

write "bu ogrenci zaten var!"

end if

else

write “hafizada yer yok!!!”

end if

case ‘ 2’:

bir\_ders🡨malloc(sizeof(struct ders))

if(bir\_ders)

then Repeat while

Write "dersin verildiği donem: "

read donem

end repeat while(donem<1 OR donem>8)

ders\_kodu🡨ders\_kodu\_belirle(dersler,donem)

bir\_ders->ders\_kodu🡨ders\_kodu

write "ders kodu:",bir\_ders->ders\_kodu

write "dersin adini giriniz:"

fflush(stdin)

gets(bir\_ders->ders\_adi)

write "dersin kredisini giriniz:"

read bir\_ders->kredi

bir\_ders->ogr\_sayisi🡨0

bir\_ders->not\_ort🡨0.00

if(dersler[ders\_kodu-1]<>NULL)

write " ",dersler[ders\_kodu-1]->ders\_kodu

bir\_ders->not\_bilgisi🡨malloc(sizeof(struct notlar))

bir\_ders->not\_bilgisi->notu🡨-1

bir\_ders->not\_bilgisi->ogr\_no🡨-1

bir\_ders->not\_bilgisi->noya\_gore\_sonraki🡨bir\_ders->not\_bilgisi

bir\_ders->not\_bilgisi->nota\_gore\_onceki🡨bir\_ders->not\_bilgisi

bir\_ders->not\_bilgisi->nota\_gore\_sonraki🡨bir\_ders->not\_bilgisi

dersler[bir\_ders->ders\_kodu-1]🡨bir\_ders

write "ekleme islemi basari ile tamamlandi..."

else

write "hafizada yeteri kadar yer yok!"

end if

case ‘ 3’:

ders\_kodu🡨ders\_kodu\_al()

kod🡨ders\_kodu/10

if (koda\_gore\_ara(dersler,ders\_kodu,kod)<>NULL)

then bir\_ders🡨koda\_gore\_ara(dersler,ders\_kodu,kod)

Repeat while

ogr\_no🡨ogr\_no\_al()

kod\_2🡨hashing(ogr\_no)

bir\_ogrenci🡨ara(ogrenciler[kod\_2],ogr\_no)

if(bir\_ogrenci<>NULL)

then liste\_dugumu🡨dersler[ders\_kodu-1]->not\_bilgisi

bir\_no🡨NULL

bir\_no🡨ders\_icinde\_ogr\_arama(liste\_dugumu,ogr\_no)

if(bir\_no=NULL)

then bir\_not🡨malloc(sizeof(struct notlar))

bir\_not->ogr\_no🡨ogr\_no

if(bir\_not)

then write "ogrencinin notunu giriniz:"

read bir\_not->notu

yeni\_not🡨bir\_not->notu

not\_guncelle(ogrenciler[kod\_2],bir\_ders,yeni\_not,ogr\_no)

ogr\_noya\_gore\_ekle(bir\_not,liste\_dugumu)

ogr\_nota\_gore\_ekle(bir\_not,liste\_dugumu)

dersler[ders\_kodu-1]->ogr\_sayisi= dersler[ders\_kodu-1]->ogr\_sayisi+1

write "ekleme islemi basariyla tamamlandi..."

end if

end if

else

write "bu ogrenci zaten bu dersi aliyor"

end if

else

write “boyle bir ogrenci yok”

write “Baska bir ogrenci var mi?”

fflush(stdin)

devam🡨getchar()

end if

while(devam 🡨'E' OR devam🡨'e')

end repeat

else

write “boyle bir ders yok ki”

end if

case ‘ 4’:

ders\_kodu🡨ders\_kodu\_al()

kod🡨ders\_kodu/10

if(koda\_gore\_ara(dersler,ders\_kodu,kod)<>NULL)

then ogr\_no🡨ogr\_no\_al()

kod\_2🡨hashing(ogr\_no)

liste\_dugumu🡨dersler[ders\_kodu-1]->not\_bilgisi

bir\_not🡨ders\_icinde\_ogr\_arama(liste\_dugumu,ogr\_no)

if(bir\_not<>NULL)

then write “ogrencinin notunu giriniz:”

read bir\_not->notu

yeni\_not🡨bir\_not->notu

not\_guncelle(ogrenciler[kod\_2],bir\_ders,yeni\_not,ogr\_no)

tasi(bir\_not,liste\_dugumu)

write “guncelleme islemi basari ile tamamlanmistir...”

end if

else write “… kodlu dersi … nolu ogrenci almiyor ki!",ders\_kodu,ogr\_no

end if

else

write “boyle bir ders yok!”

case ‘ 5’ :

ogr\_no🡨ogr\_no\_al()

kod🡨hashing(ogr\_no)

bir\_ogrenci🡨ara(ogrenciler[kod],ogr\_no)

if(bir\_ogrenci=NULL)

write “Bu ogrenci bulunmamaktadir”

else

ogrenci\_listesinden\_cikar(&ogrenciler[kod],ogr\_no)

Repeat for i🡨10 to 89

if(dersler[i+1]<>NULL)

then liste\_dugumu🡨dersler[i+1]->not\_bilgisi

notlar\_listesinden\_noya\_gore\_cikar(liste\_dugumu,ogr\_no)

notlar\_listesinden\_nota\_gore\_cikar(liste\_dugumu,ogr\_no)

end if

end repeat

write “Silme isleminiz basariyla gerceklesmistir...”

end if

case ‘6’:

ders\_kodu🡨ders\_kodu\_al()

kod🡨ders\_kodu/10

if(koda\_gore\_ara(dersler,ders\_kodu,kod)<>NULL)

then bir\_ders🡨koda\_gore\_ara(dersler,ders\_kodu,kod)

write “Ders Kodu Ders Adi Kredi Ogrenci Say Not Ort”

write “---------- ----------- ------- ----------- ---------”

write “… … … … …",bir\_ders->ders\_kodu,bir\_ders->ders\_adi,bir\_ders->kredi,bir\_ders->ogr\_sayisi,bir\_ders->not\_ort)

write “Dersi Alan Ogrenciler:”

write “Ogr No Ad Soyad Sinif Notu”

write “------- --------- ------ -------”

if(dersler[ders\_kodu-1]<>NULL)

then liste\_dugumu🡨dersler[ders\_kodu-1]->not\_bilgisi

gecici\_ogr🡨liste\_dugumu->nota\_gore\_sonraki

repeat while(gecici\_ogr<>liste\_dugumu)

ogr\_no🡨gecici\_ogr->ogr\_no

kod🡨hashing(ogr\_no)

bir\_ogrenci🡨ara(ogrenciler[kod],ogr\_no)

if(bir\_ogrenci<>NULL)

then write “… … …. …",bir\_ogrenci->ogr\_no,bir\_ogrenci->ad\_soyad,

bir\_ogrenci->sinif,gecici\_ogr->notu

end if

gecici\_ogr🡨gecici\_ogr->nota\_gore\_sonraki

end repeat

end if

end if

else

write “boyle bir ders yok ki”

end if

case ‘7’:

sayac\_1🡨0,sayac\_sinir\_alti🡨0,yüzde🡨0.00

ders\_kodu🡨ders\_kodu\_al()

kod🡨ders\_kodu/10

sayac\_sinir\_alti🡨0

if(koda\_gore\_ara(dersler,ders\_kodu,kod)<>NULL)

then write “Gormek istediginiz not barajini giriniz:”

read not\_baraji

liste\_dugumu🡨dersler[ders\_kodu-1]->not\_bilgisi

bir\_not🡨liste\_dugumu->nota\_gore\_onceki

write “Notu dusuk olan ogrenciler:”

write “Ogr No Ad Soyad Sinif Notu”

write “------- ------------ ------- -------- “

while(bir\_not<>liste\_dugumu AND bir\_not🡪notu >not\_baraji)

bir\_ogrenci->ogr\_no🡨bir\_not->ogr\_no

kod🡨hashing(bir\_ogrenci->ogr\_no)

bir\_ogr🡨ara(ogrenciler[kod],bir\_ogrenci->ogr\_no)

if(bir\_ogr<>NULL)

then write “… … … …",bir\_ogr->ogr\_no,bir\_ogr->ad\_soyad,bir\_ogr->sinif,bir\_not->notu

sayac\_sinir\_alti= sayac\_sinir\_alti+1

end if

sayac\_1=sayaç\_1+1

bir\_not🡨bir\_not->nota\_gore\_onceki

repeat while

end if

else

write “Bu Repeat whilenem ders bulunmamaktadir!”

yüzde🡨(sayac\_sinir\_alti\*100)/sayac\_1

write “notu … nin ustunde olan ogrenci sayisi ve yuzdesi: …",not\_baraji,sayac\_sinir\_alti,yuzde

case ‘8’:

sayaç🡨0,sayac\_sinir\_alti🡨0,yüzde🡨0.00

ders\_kodu🡨ders\_kodu\_al()

kod🡨ders\_kodu/10

bir\_ders🡨koda\_gore\_ara(dersler,ders\_kodu,kod)

sayac\_sinir\_alti🡨0

if(bir\_ders<>NULL)

then write “Gormek istediginiz not barajini giriniz:”

read not\_baraji

liste\_dugumu🡨dersler[ders\_kodu-1]->not\_bilgisi

bir\_not🡨liste\_dugumu->nota\_gore\_sonraki

write “Notu dusuk olan ogrenciler:”

write “Ogr No Ad Soyad Sinif Notu”

write “------- ------------ ------- -------- “

while(bir\_not<>liste\_dugumu AND bir\_not->notu <not\_baraj)

bir\_ogrenci->ogr\_no🡨bir\_not->ogr\_no

kod🡨hashing(bir\_ogrenci->ogr\_no)

bir\_ogr🡨ara(ogrenciler[kod],bir\_ogrenci->ogr\_no)

if(bir\_ogr<>NULL)

then write “… … … …",bir\_ogr->ogr\_no,bir\_ogr->ad\_soyad,bir\_ogr->sinif,bir\_not->notu)

sayac\_sinir\_alti= sayac\_sinir\_alti+1

end if

sayac=sayac+1

bir\_not🡨bir\_not->nota\_gore\_sonraki

end repeat

yüzde🡨(sayac\_sinir\_alti\*100)/sayac

write “notu …nin altinda olan ogrenci sayisi ve yuzdesi:.. , …",

not\_baraji, sayac\_sinir\_alti ,yuzde

else

write “Bu Repeat whilenem ders bulunmamaktadir!”

end if

case ‘9’:

ders\_bilgileri\_goruntule(dersler)

case ‘10’:

ogr\_no🡨ogr\_no\_al()

kod🡨hashing(ogr\_no)

bir\_ogrenci🡨ara(ogrenciler[kod],ogr\_no)

if(bir\_ogrenci)

then write “ogr no ad soyad sinif kredi say not ort “

write “ ------ -------- ----- ---------- -------”

write “… … … …",bir\_ogrenci->ogr\_no,bir\_ogrenci->ad\_soyad,

bir\_ogrenci->sinif,bir\_ogrenci->kredi,bir\_ogrenci->not\_ort

else

write “boyle bir ogrenci yok ki!”

end if

case ‘11’:

ogr\_no🡨ogr\_no\_al()

kod🡨hashing(ogr\_no)

bir\_ogrenci🡨ara(ogrenciler[kod],ogr\_no)

if(bir\_ogrenci=NULL)

then write “Bu numarada ogrenci bulunmamaktadir!!!”

else

write “ogr no ad soyad sinif kredi say not ort “

write “------- ---------- ------- ---------- ---------”

write “… … … … …",bir\_ogrenci->ogr\_no,bir\_ogrenci->ad\_soyad,

bir\_ogrenci->sinif,bir\_ogrenci->kredi,bir\_ogrenci->not\_ort

write “Aldigi Dersler:”

write “Ders Kodu Ders Adi Kredi Notu”

write “---------- ---------- -------- ------”

Repeat for i🡨1 to 89

if(dersler[i]<>NULL)

then liste\_dugumu🡨dersler[i]->not\_bilgisi

bir\_not🡨ders\_icinde\_ogr\_arama(liste\_dugumu,ogr\_no)

if(bir\_not<>NULL)

then write “… … … …",dersler[i]->ders\_kodu,dersler[i]->ders\_kodu,

dersler[i]->kredi,bir\_not->notu

end if

end repeat

end if

case ‘12’:

write “Gormek istediginiz ogrencilerin sinifini giriniz:”

read sinif

write “ogr no ad soyad sinif kredi say not ort “

write “ ------ -------- ----- ---------- -------”

ogr\_say🡨0,altmis\_say🡨0,not\_ortalama🡨0.00,toplam🡨0

Repeat for i🡨1 to100

geçici🡨ogrenciler[i]

repeat while (gecici<>NULL)

if (gecici->sinif=sinif)

then write “… … … … …",gecici->ogr\_no,gecici->ad\_soyad,gecici->sinif,

gecici->kredi,gecici->not\_ort

ogr\_say=ogr\_say+1

toplam🡨gecici->not\_ort+toplam

if(gecici->not\_ort<60)

then altmis\_say= altmis\_say+1

end if

end if

geçici🡨gecici->sonraki

end repeat

end repeat

not\_ortalama🡨 toplam/ogr\_say

write “Sinifin ogrenci sayisi:…",ogr\_say

write “Sinifin genel not ortalamasi:…",not\_ortalama

write “Donemlik agirlik not ortalamasi 60'in altinda olan ogrenci sayisi :…",altmis\_say

write “Donemlik agirlik not ortalamasi 60'in altinda olan ogrenci yuzdesi::…"

,(altmis\_say\*100)/ogr\_say)

end select

while (secenek<>13)

end repeat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure menu\_secim()

write “-------OGRENCI TAKIP SISTEMI-2---------------------”

write “1-yeni bir ogrencinin eklenmesi”

write “2-yeni bir dersin eklenmesi”

write “3-bir dersi alan ogrencinin notlarinin eklenmesi”

write “4-bir dersi alan ogrencinin notlarinin guncellenmesi”

write “5-bir ogrencinin silinmesi”

write “6-bir dersin bilgileri ve dersi alan ogrencilerin listelenmesi”

write “7-bir dersi alan ogrencilerden notu belirli bir notun ustunde olan ogrencilerin listelenmesi”

write “8-bir dersi alan ogrencilerden notu belirli bir notun altinda olan ogrencilerin listelenmesi”

write “9-bir Repeat whilenemin derslerinin listelenmesi”

write “10-bir ogrencinin bilgilerinin listelienmesi”

write “11-bir ogrencinin bilgilerinin ve aldigi derslerin listelenmesi”

write “12-bir sinifta okuyan ogrencilerin listelenmesi”

write “13-cikis”

Repeat do

write “seciminizi giriniz:”

read secim

while(secim<1 OR secim>13)

end repeat

return secim

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function hashing( no)

Sayi🡨(no-1)/100

return sayi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ogrenci\_sirali\_ekle(\*\*ilk, \*yeni)

struct ogrenci \*onceki, \*gecici

if(\*ilk=NULL)

then yeni->sonraki🡨NULL

\*ilk=yeni

else if((\*ilk)->ogr\_no > yeni->ogr\_no)

then yeni->sonraki🡨\*ilk

\*ilk🡨yeni

else

onceki🡨\*ilk

geçici🡨(\*ilk)->sonraki

repeat while(gecici<>NULL AND gecici->ogr\_no<yeni->ogr\_no)

onceki🡨gecici

geçici🡨gecici->sonraki

end repeat

yeni->sonraki🡨gecici

onceki->sonraki🡨yeni

end if

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function \*ara(\*ilk, aranan)

Repeat while(ilk<>NULL AND aranan>ilk->ogr\_no)

İlk🡨ilk->sonraki

end repeat

if((ilk=NULL) OR (aranan<ilk->ogr\_no))

then return NULL

else return ilk

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function ders\_kodu\_belirle(\*dersler[89], donem)

Repeat for i🡨 donem\*10 to (donem \*10)+8

if(dersler[i]=NULL)

then kod=i+1

return kod

end if

end repeat

return kod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function \*koda\_gore\_ara( \*dersler[89], aranan, kod)

a🡨kod\*10

repeat while(dersler[a-1]<>NULL AND dersler[a-1]->ders\_kodu<=aranan)

if (dersler[a-1]->ders\_kodu=aranan)

then return dersler[a-1]

end if

a=a+1

end repeat

return NULL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ogr\_noya\_gore\_ekle(\*bir\_not, \*listebasi)

struct notlar \*onceki

struct notlar \*gecici

onceki🡨listebasi

geçici🡨listebasi->noya\_gore\_sonraki

if(onceki=gecici)

then bir\_not->noya\_gore\_sonraki🡨listebasi

onceki🡨bir\_not

else

Repeat while(gecici<>onceki AND gecici->ogr\_no<bir\_not->ogr\_no)

onceki🡨gecici

geçici🡨gecici->noya\_gore\_sonraki

end repeat

bir\_not->noya\_gore\_sonraki🡨gecici

onceki->noya\_gore\_sonraki🡨bir\_not

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ogr\_nota\_gore\_ekle(\*bir\_not,\*listebasi)

struct notlar \*gecici

geçici🡨listebasi->nota\_gore\_sonraki

repeat while (gecici<>listebasi AND gecici->notu<bir\_not->notu)

geçici🡨gecici->nota\_gore\_sonraki

end repeat

bir\_not->nota\_gore\_onceki🡨gecici->nota\_gore\_onceki

bir\_not->nota\_gore\_sonraki🡨gecici

gecici->nota\_gore\_onceki->nota\_gore\_sonraki🡨bir\_not

gecici->nota\_gore\_onceki🡨bir\_not

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure tasi(\*bir\_not,\*listebasi)

if(bir\_not->nota\_gore\_onceki<>listebasi AND bir\_not->nota\_gore\_onceki->notu > bir\_not->notu)

then struct notlar \*geçici🡨bir\_not->nota\_gore\_onceki

repeat while(gecici<>listebasi AND gecici->notu > bir\_not->notu)

geçici🡨gecici->nota\_gore\_onceki

end repeat

bir\_not->nota\_gore\_onceki->nota\_gore\_sonraki🡨bir\_not->nota\_gore\_sonraki

bir\_not->nota\_gore\_sonraki->nota\_gore\_onceki🡨bir\_not->nota\_gore\_onceki

bir\_not->nota\_gore\_sonraki🡨gecici->nota\_gore\_sonraki

bir\_not->nota\_gore\_onceki🡨gecici

gecici->nota\_gore\_sonraki->nota\_gore\_onceki🡨bir\_not

gecici->nota\_gore\_sonraki🡨bir\_not

end if

if(bir\_not->nota\_gore\_sonraki<>listebasi AND bir\_not->notu > bir\_not->nota\_gore\_sonraki->notu)

then struct notlar \*geçici🡨bir\_not->nota\_gore\_sonraki

repeat while(gecici<>listebas iAND gecici->notu<bir\_not->notu)

geçici🡨gecici->nota\_gore\_sonraki

end repeat

bir\_not->nota\_gore\_onceki->nota\_gore\_sonraki🡨bir\_not->nota\_gore\_sonraki

bir\_not->nota\_gore\_sonraki->nota\_gore\_onceki🡨bir\_not->nota\_gore\_onceki

bir\_not->nota\_gore\_onceki🡨gecici->nota\_gore\_onceki

bir\_not->nota\_gore\_sonraki🡨gecici

gecici->nota\_gore\_onceki->nota\_gore\_sonraki🡨bir\_not

gecici->nota\_gore\_onceki🡨bir\_not

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ders\_bilgileri\_goruntule(\*dersler[89])

toplam\_kredi=0

Repeat do

write “derslerini gormek istediginiz Repeat whilenemi giriniz:”

read donem

while(donem<1 OR donem>8)

end repeat

write “Ders Kodu Ders adi Kredi Ogrenci sayisi Not ort “

write “--------- -------- ----- -------------- --------”

Repeat for i🡨 donem\*10 to (donem\*10)+8

if(dersler[i]<>NULL)

then write “… … … … …",dersler[i]->ders\_kodu,dersler[i]->ders\_adi,

dersler[i]->kredi,dersler[i]->ogr\_sayisi,dersler[i]->not\_ort)

toplam\_kredi🡨dersler[i]->kredi+toplam\_kredi

end if

end repeat

write “toplam kredi:…",toplam\_kredi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function \*sinifa\_gore\_listele(\*ogrenciler[100], aranan\_sinif,i)

struct ogrenci \*geçici🡨ogrenciler[i]

repeat while (gecici<>NULL)

if (gecici->sinif=aranan\_sinif)

then write “… … … … …",gecici->ogr\_no,gecici->ad\_soyad,gecici->sinif,gecici->kredi,

gecici->not\_ort)

end if

geçici🡨gecici->sonraki

end repeat

return gecici

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function \*ders\_icinde\_ogr\_arama( \*listebasi, aranan)

struct notlar \*gecici

geçici🡨listebasi->nota\_gore\_sonraki

repeat while(gecici<>listebasi AND gecici->ogr\_no<aranan)

geçici🡨gecici->nota\_gore\_sonraki

end repeat

if(gecici->ogr\_no=aranan)

then return gecici

else return NULL

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Function notlar\_listesinden\_nota\_gore\_cikar(\*liste\_basi, ogr\_no)

struct notlar \*gecici

geçici🡨liste\_basi->nota\_gore\_sonraki

repeat while(gecici<>liste\_basi AND gecici->ogr\_no<ogr\_no)

geçici🡨gecici->nota\_gore\_sonraki

end repeat

gecici->nota\_gore\_onceki->nota\_gore\_sonraki🡨gecici->nota\_gore\_sonraki

gecici->nota\_gore\_sonraki->nota\_gore\_onceki🡨gecici->nota\_gore\_onceki

free(gecici)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ogrenci\_listesinden\_cikar(\*\*ilk, ogr\_no)

struct ogrenci \*gecici

struct ogrenci \*onceki

if((\*ilk)->ogr\_no=ogr\_no)

then geçici🡨\*ilk

\*ilk🡨(\*ilk)->sonraki

free(gecici)

else

onceki🡨\*ilk

geçici🡨(\*ilk)->sonraki

repeat while(gecici<>NULL AND ogr\_no>gecici->ogr\_no)

onceki🡨gecici

geçici🡨gecici->sonraki

end repeat

onceki->sonraki🡨gecici->sonraki

free(gecici)

end if

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure notlar\_listesinden\_noya\_gore\_cikar(\*liste\_basi, ogr\_no)

struct notlar \*onceki, \*gecici

onceki🡨liste\_basi

geçici🡨liste\_basi->noya\_gore\_sonraki

repeat while(gecici<>liste\_basi AND gecici->ogr\_no<ogr\_no)

onceki🡨gecici

geçici🡨gecici->noya\_gore\_sonraki

end repeat

onceki->noya\_gore\_sonraki🡨gecici->noya\_gore\_sonraki

free(gecici)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure not\_guncelle(\*liste\_basi,\*bir\_ders, yeni\_not,no)

struct ogrenci \*geçici🡨liste\_basi

repeat while(gecici<>NULL AND gecici->ogr\_no<no)

geçici🡨gecici->sonraki

end repeat

if(gecici->ogr\_no=no)

then gecici->not\_ort🡨(((gecici->not\_ort)\*(gecici->kredi))+

((yeni\_not)\*(bir\_ders->kredi)))/((gecici->kredi)+(bir\_ders->kredi))

end if

bir\_ders->not\_ort🡨(((bir\_ders->not\_ort)\*(bir\_ders->kredi))+yeni\_not)/(bir\_ders->kredi)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ders\_kodu\_al()

Repeat do

write “ders kodunu giriniz:”

read ders\_kodu

while(ders\_kodu<10 OR ders\_kodu>89)

end repeat

return ders\_kodu

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure ogr\_no\_al()

Repeat do

write “ogrenci numarasi giriniz:”

read ogr\_no

while(ogr\_no<1 OR ogr\_no>10000)

end repeat

return ogr\_no

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*