

```

/*//////////////////////////////////////////////////////////////////////////
/* ////////////////////////////////////////////////////////////////////////// ALGORITMA VE PROGRAMLAMA DERSI-2 // ODEV-4 ////////////////////////////////////////////////////////////////////////// */
/*//////////////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
struct dugum{
    int no;
    char ad[11];
    char soyad[16];
    char sınıf[3];
    struct dersVerisi *ilk_ders_notu;
    struct dugum *sonraki;
    struct dugum *ad_sonraki;
};
struct dersVerisi{
    int dersKodu;
    int notu;
    struct dersVerisi *sonraki;
};
int hash1(int no)
{
    return no/20;
}
int hash2(char harf)
{
    harf=toupper(harf);
    return (int)harf-'A';
}
int menu()
{
    int secim;
    printf("\n ***** MENU *****\n\n"
        " 1 - Yeni bir ogrenci ekle\n"
        " 2 - Ders ekle\n"
        " 3 - Sinif guncelle\n"
        " 4 - Ders notu guncelle\n"
        " 5 - Ogrenci sil\n"
        " 6 - Bir ogrencinin bilgilerini listele\n"
        " 7 - Numara sirasina gore listele\n"
        " 8 - Ad sirasina gore listele\n"
        " 9 - Adinin ilk harfine gore listele\n"
        " 10- Bir siniftaki ogrencileri listele\n"
        " 11- Bir dersi almıs alan ogrencileri listele\n"
        " 12- Hic ders almamis ogrencileri listele\n"
        " 13- Cikis\n");
    printf("\n Seciminizi girin.:");
    scanf("%d",&secim);

    return secim;
}

```

```

int kontrol(int no, struct dugum **bas)
{
    struct dugum *p;
    int var=0;
    p=*bas;
    if(p==NULL)
        var=0;

    while(p!=NULL)
    {
        if((p->no)==no)
            var=1;
        p=p->sonraki;
    }

    return var;
}
void sirali_ekle(struct dugum **bas,struct dugum *yeni)
{
    struct dugum *onceki, *gecici;

    if(*bas==NULL)
    {
        yeni->sonraki=NULL;
        yeni->ilk_ders_notu=NULL;
        *bas=yeni;
    }
    else if(yeni->no < (*bas)->no)
    {
        yeni->sonraki=*bas;
        *bas=yeni;
    }
    else
    {
        gecici=(*bas)->sonraki;
        onceki=*bas;
        while(gecici!=NULL && yeni->no > gecici->no)
        {
            onceki=gecici;
            gecici=gecici->sonraki;
        }
        yeni->sonraki=gecici;
        onceki->sonraki=yeni;
    }
}
void sirali_sil(struct dugum **bas,int ogr_no)
{
    struct dugum *onceki,*gecici;

    if(*bas==NULL) printf("\n Liste bos.\n");
    else if((*bas)->no==ogr_no)
    {

```

```

    gecici=*bas;
    *bas>(*bas)->sonraki;
    free(gecici);
}
else
{
    gecici>(*bas)->sonraki;
    onceki=*bas;

    while(gecici!=NULL && ogr_no > gecici->no)
    {
        onceki=gecici;
        gecici=gecici->sonraki;
    }

    if(gecici==NULL || gecici->no < ogr_no)
        printf("\n Silinecek kayıt yok! \n");
    else
    {
        onceki->sonraki=gecici->sonraki;
        free(gecici);
    }
}
}
void sirali_ekle2(struct dersVerisi **bas,struct dersVerisi *yeni)//Buyukten kucuge ekle...
{
    struct dersVerisi *onceki, *gecici;

    if((*bas)==NULL)
    {
        yeni->sonraki=NULL;
        *bas=yeni;
    }
    else if(yeni->notu > (*bas)->notu)
    {
        yeni->sonraki=*bas;
        *bas=yeni;
    }
    else
    {
        gecici>(*bas)->sonraki;
        onceki=*bas;
        while(gecici!=NULL && yeni->notu < gecici->notu)
        {
            onceki=gecici;
            gecici=gecici->sonraki;
        }
        yeni->sonraki=gecici;
        onceki->sonraki=yeni;
    }
}
}

```

```

void sirali_sil2(struct dersVerisi **bas,int notu,int kod)
{
    struct dersVerisi *onceki,*gecici;

    if(*bas==NULL) printf("\n Liste bos.\n");
    else if((*bas)->dersKodu==kod)
    {
        gecici=*bas;
        *bas=(*bas)->sonraki;
        free(gecici);
    }
    else
    {
        gecici=(*bas)->sonraki;
        onceki=*bas;

        while(gecici!=NULL && notu > gecici->notu)
        {
            onceki=gecici;
            gecici=gecici->sonraki;
        }

    }
    if(gecici==NULL || gecici->notu < notu)
        printf("\n Silinecek kayıt yok! \n");
    else
    {
        onceki->sonraki=gecici->sonraki;
        free(gecici);
    }
}

void alfabetik_ekle(struct dugum *liste_basi, struct dugum *yeni)//dairesele ekle....
{
    struct dugum *gecici, *onceki;

    gecici=liste_basi->ad_sonraki;
    onceki=liste_basi;

    while(gecici!=NULL && gecici!=liste_basi && strcmp(gecici->ad,yeni->ad)<0)
    {
        onceki=gecici;
        gecici=gecici->ad_sonraki;
    }
    yeni->ad_sonraki=gecici;
    onceki->ad_sonraki=yeni;
}

void dersEkle(struct dugum *bas, struct dersVerisi *yeni,int no)
{
    struct dugum *p;

    p=bas;

```

```

while(p!=NULL)
{
    if(p->no==no)
    {
        if(p->ilk_ders_notu==NULL)
        {
            yeni->sonraki=NULL;
            p->ilk_ders_notu=yeni;
        }
        else
            sirali_ekle2(&(p->ilk_ders_notu),yeni);
    }
    p=p->ad_sonraki;
}

}
void alfabetik_sil(struct dugum *liste_basi,int ogr_no)//daire sel sil...
{
    struct dugum *gecici,*onceki;

    gecici=liste_basi->ad_sonraki;
    onceki=liste_basi;
    if(onceki->no==ogr_no)
    {
        onceki->ad_sonraki=gecici->ad_sonraki;
        free(gecici);
    }
    while(gecici!=NULL && gecici!=liste_basi && strcmp(gecici->ad,liste_basi->ad)<0)
    {
        onceki=gecici;
        gecici=gecici->ad_sonraki;
    }
    if(gecici->no==ogr_no)
    {
        onceki->ad_sonraki=gecici->ad_sonraki;
        free(gecici);
    }
}

void sinifGuncelle(struct dugum **bas, char *yeni,int no)
{
    struct dugum *p;
    p=*bas;
    while(p!=NULL)
    {
        if(p->no==no && p!=NULL)
        {
            strcpy(p->sinif,yeni);
            return;
        }
        p=p->sonraki;
    }
}

```

```

    }
}
int notGuncelle(struct dugum **bas, int kod,int no)
{
    int notu,aldi=0;
    struct dugum *p;
    struct dersVerisi *d;
    p=*bas;
    while(p!=NULL)
    {
        if(p->no==no && p!=NULL)
        {
            d=p->ilk_ders_notu;
            while(d!=NULL)
            {
                if(d->dersKodu==kod)
                {
                    printf("\n Yeni notu girin.: ");
                    scanf("%d",&notu);
                    d->notu=notu;
                    aldi=1;
                }
                d=d->sonraki;
            }
        }
        p=p->sonraki;
    }
    return aldi;
}
int main()
{
    struct dugum *ptr[500];
    struct dugum liste_basi[26];
    int i;
    struct dugum *birOgrenci,*p,*p1;
    struct dersVerisi *birDers,*d,*ders={NULL};
    int secim,var,no,deger,deger2,toplamDers=0,toplamNot=0,kod,aldi;
    char harf,sinif[3],cevap;

    for(i=0; i<500; i++) ptr[i]=NULL;

    for(i=0; i<26; i++)
    {
        liste_basi[i].no=-1;
        liste_basi[i].ad_sonraki=NULL;
        liste_basi[i].ilk_ders_notu=NULL;
    }
    secim=menu();
    for(;;)
    {
        switch(secim)

```

```

{
case 1:
    printf("\n Ogrenci numarasini girin.:");
    scanf("%d",&no);
    deger=hash1(no);
    var=kontrol(no,&ptr[deger]);
    if(var==1)
        printf("\nOgrenci bulunmaktadir.\n");
    else
    {
        birOgrenci=malloc(sizeof(struct dugum));
        birOgrenci->no=no;
        printf("\n Ogrencinin adini girin.:");
        fflush(stdin);
        gets(birOgrenci->ad);
        printf("\n Ogrencinin soyadini girin.:");
        fflush(stdin);
        gets(birOgrenci->soyad);
        printf("\n Ogrencinin sinifini girin.:");
        fflush(stdin);
        gets(birOgrenci->sinif);

        if(ptr[deger]==NULL)
        {
            birOgrenci->sonraki=NULL;
            birOgrenci->ilk_ders_notu=NULL;
            ptr[deger]=birOgrenci;
        }
        else
            sirali_ekle(&ptr[deger],birOgrenci);
        deger2=hash2(birOgrenci->ad[0]);
        if(liste_basi[deger2].no==-1)
        {
            birOgrenci->ad_sonraki=NULL;
            liste_basi[deger2]=*birOgrenci;
            liste_basi[deger2].ad_sonraki=birOgrenci;
        }
        else
            alfabetik_ekle(&liste_basi[deger2],birOgrenci);
        printf("\n Ekleme islemi gerceklesti.\n");
    }

    var=0;
    break;

case 2:
    printf("\n Ogrenci numarasini girin.:");
    scanf("%d",&no);
    deger=hash1(no);
    var=kontrol(no,&ptr[deger]);
    if(var==0)
        printf("\n Ogrenci bulunmamaktadir.\n");

```

```

else
{
do{
birDers=malloc(sizeof(struct dersVerisi));
printf("\n Ders kodunu girin:");
scanf("%d",&birDers->dersKodu);
printf("\n Ders notunu girin:");
scanf("%d",&birDers->notu);
p=ptr[deger];
if(p->no==no)
{
deger2=hash2(p->ad[0]);
if(p->ilk_ders_notu==NULL)
{
birDers->sonraki=NULL;
p->ilk_ders_notu=birDers;
}
else
{
sirali_ekle2(&p->ilk_ders_notu,birDers);
}
}
while(p->no!=no && p!=NULL)
{
if(p->no==no)
{
deger2=hash2(p->ad[0]);
if(p->ilk_ders_notu==NULL)
p->ilk_ders_notu=birDers;
else
{
ders=p->ilk_ders_notu;
sirali_ekle2(&ders,birDers);
}
}
p=p->sonraki;
}
dersEkle(&liste_basi[deger2],birDers,no);

printf("\n Ekleme islemi gerceklesti.\n"
"\n Eklemeye devam etmek ister misiniz? (E/e-H/h)");
fflush(stdin);
cevap=getchar();
}while(cevap=='E' || cevap=='e');
}
break;
case 3:
printf("\n Ogrenci numarasini girin:");
scanf("%d",&no);
deger=hash1(no);
var=kontrol(no,&ptr[deger]);
if(var==0)

```



```

    printf("\n Ogresci bulunmamaktadır.\n");
else
{
    printf("\n Yeni sinifi girin.:");
    fflush(stdin);
    gets(sinif);
    sinifGuncelle(&ptr[deger],sinif,no);
    printf("\n Guncelleme islemi gerceklestirildi. \n");
}
break;
case 4:
    printf("\n Ogresci numarasini girin.:");
    scanf("%d",&no);
    deger=hash1(no);
    var=kontrol(no,&ptr[deger]);
    if(var==0)
        printf("\n Ogresci bulunmamaktadır.\n");
    else
    {
        printf("\n Ders kodunu girin.:");
        scanf("%d",&kod);
        aldi=notGuncelle(&ptr[deger],kod,no);
        if(aldi==1)
            printf("\n Guncelleme islemi gerceklestirildi. \n");
        else
            printf("\n Ogresci bu dersi daha once almadigindan guncelleme yapilamadi.\n");
    }
    break;
case 5:
    printf("\n Ogresci numarasini girin.:");
    scanf("%d",&no);
    deger=hash1(no);
    var=kontrol(no,&ptr[deger]);
    if(var==0)
        printf("\n Ogresci bulunmamaktadır.\n");
    else
    {
        p=ptr[deger];
        p1=ptr[deger];
        sirali_sil(&ptr[deger],no);
        while(p!=NULL)
        {
            if(p->no==no)
            {
                d=p->ilk_ders_notu;
                while(d!=NULL)
                {
                    sirali_sil2(&d,d->notu,d->dersKodu);
                    d=d->sonraki;
                }
            }
            p=p->sonraki;
        }
    }

```

```

    }
    deger2=hash2(p1->ad[0]);
    alfabetik_sil(&liste_basi[deger2],no);
    printf("\n Silme islemi gerceklesti.\n");
}
break;
case 6:
    printf("\n Ogrenci numarasini girin.:");
    scanf("%d",&no);
    deger=hash1(no);
    var=kontrol(no,&ptr[deger]);
    toplamDers=toplamNot=0;
    if(var==0)
        printf("\n Ogrenci bulunmamaktadır.\n");
    else
    {
        printf(" Ogr No Ad      Soyad      Sinif\n"
            " -----\n");
        p=ptr[deger];
        while(p!=NULL)
        {
            if((p->no)==no)
            {
                printf(" %-7d%-11s%-16s%-5s\n",p->no,p->ad,p->soyad,p->sinif);
                ders=p->ilk_ders_notu;
            }
            p=p->sonraki;
        }
        printf("\n Almis Oldugu Dersler:\n"
            " Ders Kodu Notu\n"
            " -----\n");
        if(ders==NULL)
            printf("Ogrencinin aldigi ders bulunmamaktadır.\n");
        else
        {
            d=ders;
            while(d!=NULL)
            {
                printf(" %-9d %-4d\n",d->dersKodu,d->notu);
                toplamDers=toplamDers+1;
                toplamNot=(d->notu)+toplamNot;
                d=d->sonraki;
            }
            printf("\n Aritmetik not ortalamasi.: %.2f", (float)toplamNot/(float)toplamDers);
        }
    }
    break;
case 7:
    printf(" Ogr No Ad      Soyad      Sinif\n"
        " -----\n");
    for(i=0; i<500; i++)
    {

```

```

    p=ptr[i];
    while(p!=NULL)
    {
        printf(" %-7d%-11s%-16s%-5s\n",p->no,p->ad,p->soyad,p->sinif);
        p=p->sonraki;
    }
}
break;
case 8:
printf(" Ad      Soyad      Ogr No Sinif\n"
" -----\n");
for(i=0; i<26; i++)
{
    if(liste_basi[i].no!=-1)
    {
        p=liste_basi[i].ad_sonraki;
        if(p==&liste_basi[i])
            printf(" %-11s%-16s%-7d%-5s\n",p->ad,p->soyad,p->no,p->sinif);

        while(p!=&liste_basi[i] && p!=NULL)
        {
            printf(" %-11s%-16s%-7d%-5s\n",p->ad,p->soyad,p->no,p->sinif);
            p=p->ad_sonraki;
        }
    }
}
break;
case 9:
printf("\n Harfi girin:");
fflush(stdin);
harf=getchar();
deger2=hash2(harf);
printf("\n Ad      Soyad      Ogr No Sinif\n"
" -----\n");
if(liste_basi[deger2].no!=-1)
{
    p=liste_basi[deger2].ad_sonraki;
    if(p==&liste_basi[deger2])
        printf(" %-11s%-16s%-7d%-5s\n",p->ad,p->soyad,p->no,p->sinif);

    while(p!=&liste_basi[deger2] && p!=NULL && toupper(p->ad[0])==toupper(harf))
    {
        printf(" %-11s%-16s%-7d%-5s\n",p->ad,p->soyad,p->no,p->sinif);
        p=p->ad_sonraki;
    }
}
break;
case 10:
printf("\n Sinifi girin:");
fflush(stdin);
gets(sinif);
printf("\n Ad      Soyad      Ogr No Sinif Ders Sayisi Not Ort\n"

```

```

        " -----\n");
for(i=0; i<500; i++)
{
    p=ptr[i];
    while(p!=NULL)
    {
        toplamDers=0;
        toplamNot=0;
        if(strcmp(p->sinif,sinif)==0)
        {
            d=p->ilk_ders_notu;
            while(d!=NULL)
            {
                toplamDers=toplamDers+1;
                toplamNot=toplamNot+(d->notu);
                d=d->sonraki;
            }
            printf(" %-11s%-16s%-7d%-5s %-12d%-8.2f\n",p->ad,p->soyad,p->no,p-
>sinif,toplamDers,(float)toplamNot/(float)toplamDers);
        }
        p=p->sonraki;
    }
}
break;
case 11:
    printf("\n Ders kodunu girin.: ");
    scanf("%d",&kod);
    printf("\n Ogr No Ad Soyad Notu\n"
        " -----\n");
    toplamDers=0;
    toplamNot=0;
    for(i=0; i<500; i++)
    {
        p=ptr[i];
        while(p!=NULL)
        {
            d=p->ilk_ders_notu;
            while(d!=NULL)
            {
                if(d->dersKodu==kod)
                {
                    toplamDers=toplamDers+1;
                    toplamNot=toplamNot+(d->notu);
                    printf(" %-7d%-11s%-16s%-4d\n",p->no,p->ad,p->soyad,d->notu);
                }
                d=d->sonraki;
            }
        }
        p=p->sonraki;
    }
}
printf("\n Dersin genel not ortalamasi.: %.2f", (float)toplamNot/(float)toplamDers);
break;

```

```

case 12:
    printf("\n Ad      Soyad      Ogr No\n"
           "-----\n");
    for(i=0; i<26; i++)
    {
        p=liste_basi[i].ad_sonraki;
        if(liste_basi[i].no!=-1)
        {
            if(liste_basi[i].ilk_ders_notu==NULL && p==&liste_basi[i])
                printf(" %-11s%-16s%-7d\n",liste_basi[i].ad,liste_basi[i].soyad,liste_basi[i].no);

            while(p!=&liste_basi[i] && p!=NULL)
            {
                if(p->ilk_ders_notu==NULL)
                    printf(" %-11s%-16s%-7d\n",p->ad,p->soyad,p->no);
                p=p->ad_sonraki;
            }
        }
    }
    break;
case 13:
    exit(0);
    break;

}
printf("\n Menuye donmek icin herhangi bir tusa basin.:");
fflush(stdin);
cevap=getchar();
system("cls");
secim=menu();
}
return 0;
}

```