```
/*
ÖDEV 1
Öğreci No: 1030515650
Öğrenci Adı: Sait ORHAN
Ders Kodu:BZ205
Dosya İsmi: Odev_1
*/
#include <iostream>
#include <stdlib.h>
#include <string.h>

#define BILGISAYAR 10
#define ELEKTRIK 11
#define CEVRE 12
#define HARITA 13
#define INSAAT 14
#define MAKINA 15
#define MEKATRONIK 16
#define BIYOMEDIKAL 17
#define ENDUSTRI 18
#define YAZILIM 19

using namespace std;
void HerIkiDersiAlanlar(void);
void SadeceProgramlamaDersiAlanlar(void);
void EnAzBiriniAlan_Bilgisayar(void);
void Menu(void);
void SecimiYap(const int secim);

class Veriler
{
public:
    int numara;
    char isim[15];
    char soyisim[15];
    Veriler *sonraki;
    Veriler *onceki;
    void BolumBelirle(void);
    void BolumYaz(void);
private:
    char bolum[15];
};

class Liste
{
private:
    Veriler *ListeBasi;
    Veriler *ListeSonu;

public:
    Liste()
    {
        ListeBasi = NULL;
        ListeSonu = NULL;
    }
    ~Liste();
    Veriler* ListeBasiDondur()  { return ListeBasi; }
    Veriler* ListeSonuDondur()  { return ListeSonu; }
    bool ListeBosMu();
    void ListeyeEkle(void);
    void ListeyeEkle(Veriler *Eklenecek);
    bool OgrenciBul(const int ArananNumara);
    void VeriSil(const int SilinecekNumara);
    void ListeyiGoruntele(void);
};

class YiginVerileri
{
    public:
    int numara;
    char isim[15];
    char soyisim[15];
    YiginVerileri *sonraki;
    void BolumBelirle(void);
    void BolumYaz(void);
```

```
    private:
    char bolum[15];
};

class Yigin
{
    public:
    Yigin()
    {
        YiginTepesi = NULL;
    }
    ~Yigin();
    bool YiginBosMu(void)   { return YiginTepesi == NULL; }
    int ElemanSayisi(void);
    void TepeyeElemanEkle(void);
    bool YiginiTara(const int ArananNumara);
    void TepedenElemanSil(void);
    void TepeElemaniniGoster(void);
    void TumElemanlariYaz(void);
    YiginVerileri* YiginTepesiDondur()  { return YiginTepesi; }
    private:
    YiginVerileri *YiginTepesi;
};

Liste *VeriTabani = new Liste();
Liste *Tekler = new Liste();
Liste *Ciftler = new Liste();
Yigin *Programlama = new Yigin();


int main()
{
    system("color F0");
    Menu();
    int secim;
    cin >> secim;
    SecimiYap(secim);

    cout <<"\a\n" << endl;
    return 0;
}

// Veriler Metodları
void Veriler::BolumBelirle(void)
{
    switch(numara/1000)
    {
        case BILGISAYAR:
            strcpy(bolum, "Bilgisayar");    break;
        case ELEKTRIK:
            strcpy(bolum, "Elektrik");      break;
        case CEVRE:
            strcpy(bolum, "Cevre");    break;
        case HARITA:
            strcpy(bolum, "Harita");      break;
        case INSAAT:
            strcpy(bolum, "insaat");    break;
        case MAKINA:
            strcpy(bolum, "Makina");      break;
        case MEKATRONIK:
            strcpy(bolum, "Mekatronik");    break;
        case BIYOMEDIKAL:
            strcpy(bolum, "Biyomedikal");    break;
        case ENDUSTRI:
            strcpy(bolum, "Endustri");    break;
        case YAZILIM:
            strcpy(bolum, "Yazilim");    break;
    }
}

void Veriler::BolumYaz(void)
{
    cout << bolum << " Muhendisligi ";
}
```

```
//Liste Metotları
bool Liste::ListeBosMu(void)
{
    return ListeBasi==NULL && ListeSonu==NULL;
};

void Liste::ListeyeEkle(void)
{
    int secim=1;
    while(secim==1)
        {
            Veriler *Eklenecek = new Veriler();
            if(Eklenecek)
                {
                    cout << "Yeni Ogrencinin Numarasini Giriniz: ";
                    cin >> Eklenecek->numara;
                    if(OgrenciBul(Eklenecek->numara))
                    {
                        cout << Eklenecek->numara << " ile daha once kayit alinmistir. Numarayi Kontrol ↙
    ediniz!\n";
                        continue;
                    }
                    Eklenecek->BolumBelirle();
                    for(;Eklenecek->numara/1000 >= 20 || Eklenecek->numara/1000 < 10;)
                        {
                        cout << "Hatali Numara Girisi! Numarayi Tekrar Giriniz.\n";
                        cout << "Yeni Ogrencinin Numarasini Giriniz: ";
                        cin >> Eklenecek->numara;
                            if(OgrenciBul(Eklenecek->numara))
                                {
                                    cout << Eklenecek->numara << " ile daha once kayit alinmistir.       ↙
    Numarayi Kontrol ediniz!\n";
                                    Eklenecek->numara = 0;
                                    continue;
                                }
                        }
                        Eklenecek->BolumBelirle();
                    cout << "Yeni Ogrencinin Adi ve Soyadini Giriniz: ";
                    cin >> Eklenecek->isim; cin >> Eklenecek->soyisim;
                    Veriler *SiniflandirmadaKullan = new Veriler();
                    SiniflandirmadaKullan->numara = Eklenecek->numara;
                    SiniflandirmadaKullan->BolumBelirle();
                    strcpy(SiniflandirmadaKullan->isim, Eklenecek->isim);
                    strcpy(SiniflandirmadaKullan->soyisim, Eklenecek->soyisim);
                    if(SiniflandirmadaKullan->numara%2 == 0)
                        Ciftler->ListeyeEkle(SiniflandirmadaKullan);
                    else
                        Tekler->ListeyeEkle(SiniflandirmadaKullan);
                    //Eklenecek Eleman Listenin İlk Elamanı mı?
                    if(ListeSonu == NULL)
                    {
                        ListeBasi = Eklenecek;
                        ListeSonu = Eklenecek;
                        ListeSonu->onceki = NULL;
                        ListeSonu->sonraki = NULL;
                    }
                    //Eklenecek Eleman Listenin Başına mı Eklenecek?
                    else if(ListeBasi->numara > Eklenecek->numara)
                    {
                        ListeBasi->onceki = Eklenecek;
                        Eklenecek->sonraki = ListeBasi;
                        ListeBasi = Eklenecek;
                        ListeBasi->onceki = NULL;
                    }
                    //Eklenecek Eleman Listenin Sonuna mı Eklenecek?
                    else if(Eklenecek->numara > ListeSonu->numara)
                    {
                        ListeSonu->sonraki = Eklenecek;
                        Eklenecek->onceki = ListeSonu;
                        ListeSonu = Eklenecek;
                        ListeSonu->sonraki = NULL;
                    }
                    //Eklenecek Eleman Araya mı Girecek?
                    else
                    {
```

```
                        Veriler *Bakilan;
                        Veriler *Bakilandan_Onceki;
                        Bakilan = ListeBasi;
                        while(Eklenecek->numara > Bakilan->numara)
                        {
                            Bakilandan_Onceki=Bakilan;
                            Bakilan = Bakilan->sonraki;
                        }
                        Eklenecek->sonraki = Bakilan;
                        Eklenecek->onceki = Bakilandan_Onceki;
                        Bakilandan_Onceki->sonraki = Eklenecek;
                        Bakilan->onceki = Eklenecek;
                    }

                }
                cout << "\nYeni Ogrenci Girisi Basarili. Yeni Giris Icin 1 e basiniz.\n"
                "Cikmak Icin 1 Disinda Rakam Giriniz.Seciminiz: ";
                cin>>secim;
                cout<<endl;
        }
}

void Liste::ListeyeEkle(Veriler *Eklenecek)
{
    if((Eklenecek->numara)/1000 >=10 && (Eklenecek->numara)/1000 < 20)
        Eklenecek->BolumBelirle();
    else
        {
            cout << Eklenecek->numara << " numarali ogrenci kayit yapamamaktadir.\n";
            return;
        }
    //Eklenecek Eleman Listenin İlk Elamanı mı?
    if(ListeSonu == NULL)
      {
       ListeBasi = Eklenecek;
       ListeSonu = Eklenecek;
       ListeSonu->onceki = NULL;
       ListeSonu->sonraki = NULL;
       cout << Eklenecek->numara << " numarali ogrenci eklendi.\n";
      }
    //Eklenecek Eleman Listenin Başına mı Eklenecek?
    else if(ListeBasi->numara > Eklenecek->numara)
           {
            ListeBasi->onceki = Eklenecek;
            Eklenecek->sonraki = ListeBasi;
            ListeBasi = Eklenecek;
            ListeBasi->onceki = NULL;
            cout << Eklenecek->numara << " numarali ogrenci eklendi.\n";
           }
    //Eklenecek Eleman Listenin Sonuna mı Eklenecek?
    else if(Eklenecek->numara > ListeSonu->numara)
           {
            ListeSonu->sonraki = Eklenecek;
            Eklenecek->onceki = ListeSonu;
            ListeSonu = Eklenecek;
            ListeSonu->sonraki = NULL;
            cout << Eklenecek->numara << " numarali ogrenci eklendi.\n";
           }
    //Eklenecek Eleman Araya mı Girecek?
    else
        {
          Veriler *Bakilan;
          Veriler *Bakilandan_Onceki;
          Bakilan = ListeBasi;
          while(Eklenecek->numara > Bakilan->numara)
               {
                Bakilandan_Onceki=Bakilan;
                Bakilan = Bakilan->sonraki;
               }
          Eklenecek->sonraki = Bakilan;
          Eklenecek->onceki = Bakilandan_Onceki;
          Bakilandan_Onceki->sonraki = Eklenecek;
          Bakilan->onceki = Eklenecek;
          cout << Eklenecek->numara << " numarali ogrenci eklendi.\n";
```

```
        }
}

bool Liste::OgrenciBul(const int ArananNumara)
{
    //Liste Boş mu?
    if(ListeSonu==NULL)
        return false;
    //Liste Numara Sıralı Olduğundan ArananNumara ListeBasindaki Numaradan Küçükse Listede Yoktur.
    else if(ListeBasi->numara > ArananNumara)
        return false;
    //ArananNumara Listenin Sonundaki Numaradan Büyükse Listede Yoktur.
    else if(ListeSonu->numara < ArananNumara)
        return false;
    Veriler *Bakilan = new Veriler();
    Bakilan = ListeBasi;
    while(Bakilan != NULL && Bakilan->numara != ArananNumara)
        Bakilan = Bakilan->sonraki;
    if(Bakilan == NULL)
        return false;
    else if(Bakilan->numara == ArananNumara)
        return true;
}

void Liste::VeriSil(const int SilinecekNumara)
{
    Veriler *Silinecek;
    //Listede Eleman Var mı?
    if(ListeSonu == NULL)
    {
        cout << "Liste Bos!\n";
        return;
    }
    //Listede Tek Eleman Varsa ve Silinecek Eleman ise
    else if(ListeBasi == ListeSonu && ListeBasi->numara == SilinecekNumara)
    {
        ListeBasi=NULL;
        ListeSonu=NULL;
        cout << SilinecekNumara << " numarali ogrenci listeden silindi.\nListe bosaldi!\n";
        return;
    }
    //Silinecek Eleman ilk Eleman mı
    else if(ListeBasi->numara == SilinecekNumara)
    {
        Silinecek=ListeBasi;
        ListeBasi=ListeBasi->sonraki;
        ListeBasi->onceki=NULL;
        delete Silinecek;
        cout << SilinecekNumara << " numarali ogrenci listeden silindi.\n";
        return;
    }
    //Silinecek Eleman Son Eleman mı
    else if(ListeSonu->numara == SilinecekNumara)
    {
        Silinecek=ListeSonu;
        ListeSonu=ListeSonu->onceki;
        ListeSonu->sonraki=NULL;
        delete Silinecek;
        cout << SilinecekNumara << " numarali ogrenci listeden silindi.\n";
        return;
    }
    //Silinecek Eleman için Araya Bak
    else
    {
        Silinecek = ListeBasi;
        while(Silinecek != NULL && Silinecek->numara != SilinecekNumara)
            Silinecek = Silinecek->sonraki;
        if(Silinecek==NULL)
            cout << SilinecekNumara << " numarali ogrenci listede bulunamadi.\n";
        else if(Silinecek->numara == SilinecekNumara)
                {
                    (Silinecek->onceki)->sonraki = Silinecek->sonraki;
                    (Silinecek->sonraki)->onceki = Silinecek->onceki;
                    delete Silinecek;
                    cout << SilinecekNumara << " numarali ogrenci listeden silindi.\n";
```

```
                    return;
                }
    }
}

void Liste::ListeyiGoruntele(void)
{
    if(ListeSonu==NULL)
        cout << "Liste bos.\n";
    else
    {
        Veriler *Bakilan;
        Bakilan = ListeBasi;
        cout << "Listedeki Veriler\n";
        while(Bakilan!=NULL)
        {
            cout << Bakilan->numara;
            cout<<" ";
            Bakilan->BolumYaz();
            cout.width(3);
            cout << "  " << Bakilan->isim << " " << Bakilan->soyisim << endl;
            Bakilan=Bakilan->sonraki;

        }
    }
}

//YiginVerileri Metodları
void YiginVerileri::BolumBelirle(void)
{
    switch(numara/1000)
    {
        case BILGISAYAR:
            strcpy(bolum, "Bilgisayar");    break;
        case ELEKTRIK:
            strcpy(bolum, "Elektrik");      break;
        case CEVRE:
            strcpy(bolum, "Cevre");    break;
        case HARITA:
            strcpy(bolum, "Harita");      break;
        case INSAAT:
            strcpy(bolum, "insaat");    break;
        case MAKINA:
            strcpy(bolum, "Makina");      break;
        case MEKATRONIK:
            strcpy(bolum, "Mekatronik");    break;
        case BIYOMEDIKAL:
            strcpy(bolum, "Biyomedikal");    break;
        case ENDUSTRI:
            strcpy(bolum, "Endustri");    break;
        case YAZILIM:
            strcpy(bolum, "Yazilim");    break;
    }
}

void YiginVerileri::BolumYaz(void)
{
    cout << bolum << " Muhendisligi ";
}

//Yigin Metodları
void Yigin::TepeyeElemanEkle(void)
{
    int secim=1;
    while(secim==1)
        {
            YiginVerileri *Eklenecek = new YiginVerileri();
            if(Eklenecek)
                {
                    cout << "Yeni Ogrencinin Numarasini Giriniz: ";
                    cin >> Eklenecek->numara;
                    if(YiginiTara(Eklenecek->numara))
                    {
                        cout << Eklenecek->numara << " ile daha once kayit alinmistir. Numarayi Kontrol ↙
    ediniz!\n";
```

```
                        continue;
                    }
                    Eklenecek->BolumBelirle();
                    for(;Eklenecek->numara/1000 >= 20 || Eklenecek->numara/1000 < 10;)
                        {
                        cout << "Hatali Numara Girisi! Numarayi Tekrar Giriniz.\n";
                        cout << "Yeni Ogrencinin Numarasini Giriniz: ";
                        cin >> Eklenecek->numara;
                        if(YiginiTara(Eklenecek->numara))
                                {
                                    cout << Eklenecek->numara << " ile daha once kayit alinmistir.        ↙
    Numarayi Kontrol ediniz!\n";
                                    Eklenecek->numara = 0;
                                    continue;
                                }
                        }
                        Eklenecek->BolumBelirle();
                    cout << "Yeni Ogrencinin Adi ve Soyadini Giriniz: ";
                    cin >> Eklenecek->isim; cin >> Eklenecek->soyisim;
                }
            //Eklenecek Eleman Yığınının ilk elemanı ise
            if(YiginTepesi==NULL)
            {
                YiginTepesi = Eklenecek;
                YiginTepesi->sonraki = NULL;
            }
            else
            {
                Eklenecek->sonraki = YiginTepesi;
                YiginTepesi = Eklenecek;
            }
            cout << "\nYeni Ogrenci Girisi Basarili. Yeni Giris Icin 1 e basiniz.\n"
                "Cikmak Icin 1 Disinda Rakam Giriniz.Seciminiz: ";
            cin>>secim;
            cout<<endl;
        }
}

void Yigin::TepedenElemanSil(void)
{
    //Yığınn Boş ise
    if(YiginTepesi==NULL)
        cout << "Hata! Yigin Bos.\n";
    else
    {
        YiginVerileri *Silinecek;
        Silinecek = YiginTepesi;
        YiginTepesi = YiginTepesi->sonraki;
        delete Silinecek;
        cout << "Tepeden Eleman Silindi\n";
        if(YiginTepesi == NULL)
            cout << "Silme sonucunda yiginda eleman kalmadi.\n";
    }
}

void Yigin::TepeElemaniniGoster(void)
{
    cout << "Yiginin tepesindeki eleman:\n";
    cout << YiginTepesi->numara << " ";
    YiginTepesi->BolumYaz();
    cout << " " << YiginTepesi->isim << " " << YiginTepesi->soyisim << endl;
}

int Yigin::ElemanSayisi(void)
{
    int eleman=0;
    YiginVerileri *Bakilan;
    Bakilan = YiginTepesi;
    while(Bakilan!=NULL)
    {
        Bakilan = Bakilan->sonraki;
        eleman++;
    }
    return eleman;
}
```

```
void Yigin::TumElemanlariYaz(void)
{
    cout << "Yigindaki Elemanlar:\n";
    YiginVerileri *Bakilan;
    Bakilan=YiginTepesi;
    while(Bakilan != NULL)
    {
        cout << Bakilan->numara << " ";
        Bakilan->BolumYaz();
        cout << Bakilan->isim << " " << Bakilan->soyisim << endl;
        Bakilan = Bakilan->sonraki;
    }
}

bool Yigin::YiginiTara(const int ArananNumara)
{
    YiginVerileri *Bakilan;
    Bakilan = YiginTepesi;
    while(Bakilan!=NULL && Bakilan->numara != ArananNumara)
            Bakilan = Bakilan->sonraki;
    if(Bakilan == NULL)
        return false;
    else if(Bakilan->numara == ArananNumara)
        return true;
}


void HerIkiDersiAlanlar(void)
{
    YiginVerileri *YiginaBak;
    Veriler *ListeyeBak;
    YiginaBak = Programlama->YiginTepesiDondur();
    ListeyeBak = VeriTabani->ListeBasiDondur();
    while(YiginaBak != NULL)
    {
        if(VeriTabani->OgrenciBul(YiginaBak->numara))
        {
            cout << YiginaBak->numara << " ";
            YiginaBak->BolumYaz();
            cout << YiginaBak->isim << " " << YiginaBak->soyisim << endl;
        }
        YiginaBak = YiginaBak->sonraki;
    }
}

void SadeceProgramlamaDersiAlanlar(void)
{
    YiginVerileri *YiginaBak;
    Veriler *ListeyeBak;
    YiginaBak = Programlama->YiginTepesiDondur();
    ListeyeBak = VeriTabani->ListeBasiDondur();
    while(YiginaBak != NULL)
    {
        if(!VeriTabani->OgrenciBul(YiginaBak->numara))
        {
            cout << YiginaBak->numara << " ";
            YiginaBak->BolumYaz();
            cout << YiginaBak->isim << " " << YiginaBak->soyisim << endl;
        }
        YiginaBak = YiginaBak->sonraki;
    }
}

void EnAzBiriniAlan_Bilgisayar(void)
{
    YiginVerileri *YiginaBak;
    Veriler *ListeyeBak;
    YiginaBak = Programlama->YiginTepesiDondur();
    ListeyeBak = VeriTabani->ListeBasiDondur();
    while(ListeyeBak != NULL)
    {
        if((ListeyeBak->numara)/1000 == BILGISAYAR)
        {
            cout << ListeyeBak->numara << " ";
```

```
            ListeyeBak->BolumYaz();
            cout << ListeyeBak->isim << " " << ListeyeBak->soyisim << endl;
        }
        ListeyeBak = ListeyeBak->sonraki;
    }

    while(YiginaBak !=NULL)
    {
        if( !VeriTabani->OgrenciBul(YiginaBak->numara) && YiginaBak->numara/1000 == BILGISAYAR)
        {
            cout << YiginaBak->numara << " ";
            YiginaBak->BolumYaz();
            cout << YiginaBak->isim << " " << YiginaBak->soyisim << endl;
        }
        YiginaBak = YiginaBak->sonraki;
    }
}

void Menu(void)
{
    cout << "Seciminizi Kod Numarasina Gore Belirtiniz\n"
            "----------------------------------------\n"
            "1. Programlama ve Veri Tabani Dersi Alan Ogrenciler\n"
            "2. Sadece Programlama Dersi Alan Ogrenciler\n"
            "3. Veri Tabani Dersi A (Tek Numaralilar) Grubu\n"
            "4. Veri Tabani Dersi B (Cift Numaralilar) Grubu\n"
            "5. Derslerden En az Birini Alip Bilgisayar Muh. Ogrencisi Olanlar\n"
            "6. Programlama Dersini Alan Ogrencilerin Sayisi\n"
            "Seciminiz: ";
}

void SecimiYap(const int secim)
{
    switch(secim)
    {
        case 1:
            HerIkiDersiAlanlar();   break;
        case 2:
            SadeceProgramlamaDersiAlanlar(); break;
        case 3:
            Tekler->ListeyiGoruntele(); break;
        case 4:
            Ciftler->ListeyiGoruntele(); break;
        case 5:
            EnAzBiriniAlan_Bilgisayar(); break;
        case 6:
            cout << "Programlama Dersi Alanlarin Sayisi: " << Programlama->ElemanSayisi(); break;
        default:
            cout << "Gecersiz Secim!\n";
    }
}
```