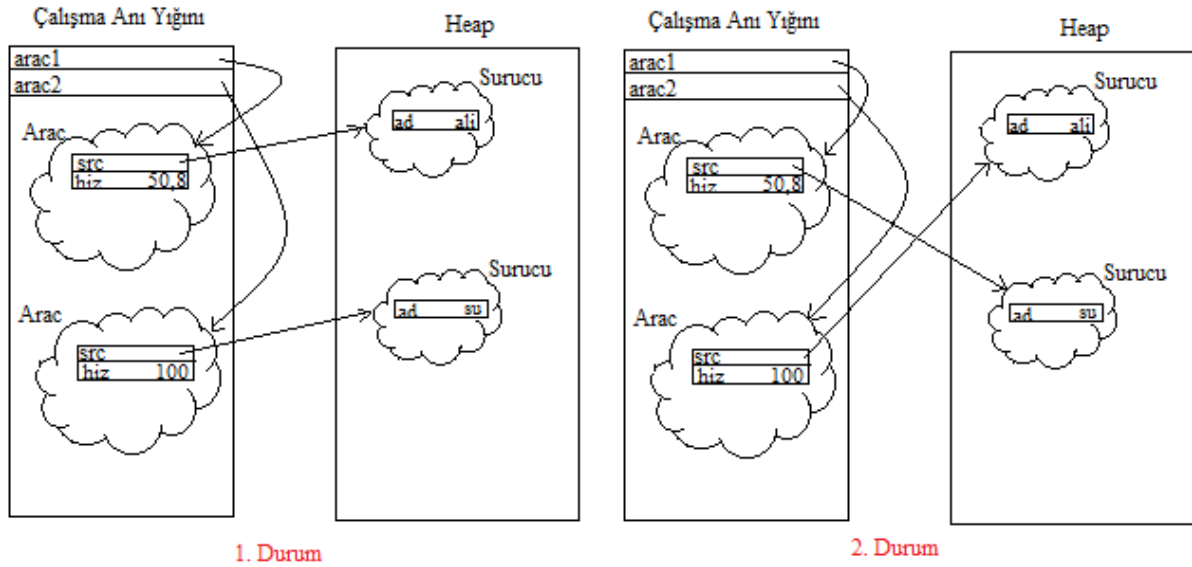


1.) Aşağıdaki 1. Durumu 2. Duruma getirecek kodu yazınız. Sınıfın detayları yazılmalıdır. (50 p)



Cevap:

```
#include <iostream>
using namespace std;

class Surucu{
public:
    string ad;
    Surucu(string isim) : ad(isim) { }
};

class Arac{
public:
    Surucu *src;
    float hiz;
    Arac(string isim, float hz){
        src = new Surucu(isim);
        hiz = hz;
    }
    ~Arac(){
        delete src;
    }
};

int main(){
    Arac arc1("ali",50.8);
    Arac arc2("su",100);
    Arac *arac1 = &arc1;
    Arac *arac2 = &arc2;

    Surucu *tmp = arac1->src;
    arac1->src = arac2->src;
    arac2->src = tmp;

    return 0;
}
```

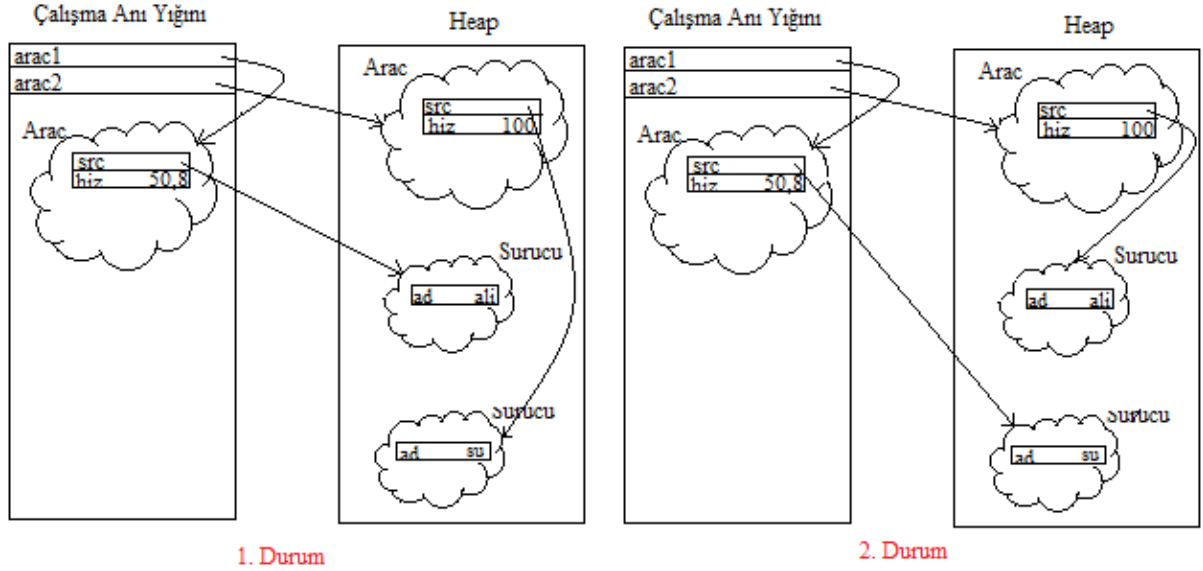
- 2.) Çift yönlü bağlı listede listeyi tersten yazdıracak metodu yazınız. Yazarken bir düğümü yazıp, bir düğümü atlayarak ilerlemelidir. Kullanılacak bazı metotların yazılmış olduğu varsayılacaktır. (50 p)

Cevap

```
void reverse(){
    if(this->isEmpty())
        cout<<"Liste bos."<<endl;
    else
    {
        Node<Nesne> *dugum=header;
        while(dugum->next != NULL) dugum=dugum->next;

        for(int i=0; dugum->prev != NULL; dugum = dugum->prev,i++){
            if(i%2==0) cout<<dugum->eleman<<" ";
        }
    }
}
```

- 1.) Aşağıdaki 1. Durumu 2. Duruma getirecek kodu yazınız. Sınıfın detayları yazılmalıdır. (50 p)



```
#include <iostream>
using namespace std;
class Surucu{
public:
    string ad;
    Surucu(string isim) : ad(isim) { }
};
class Arac{
public:
    Surucu *src;
    float hiz;
    Arac(string isim, float hz){
        src = new Surucu(isim);
        hiz = hz;
    }
    ~Arac(){
        delete src;
    }
}
```

```

    }
};
int main(){
    Arac arc1("ali",50.8);
    Arac *arac1 = &arc1;
    Arac *arac2 = new Arac("su",100);

    Surucu *tmp = arac1->src;
    arac1->src = arac2->src;
    arac2->src = tmp;

    delete arac2;
    return 0;
}

```

- 2.) Tek yönlü bağlı listede listeyi tersten yazdıracak metodu yazınız. Yazarken bir düğümü yazıp, bir düğümü atlayarak ilerlemelidir. Kullanılacak bazı metotların yazılmış olduğu varsayılacaktır. (50 p)

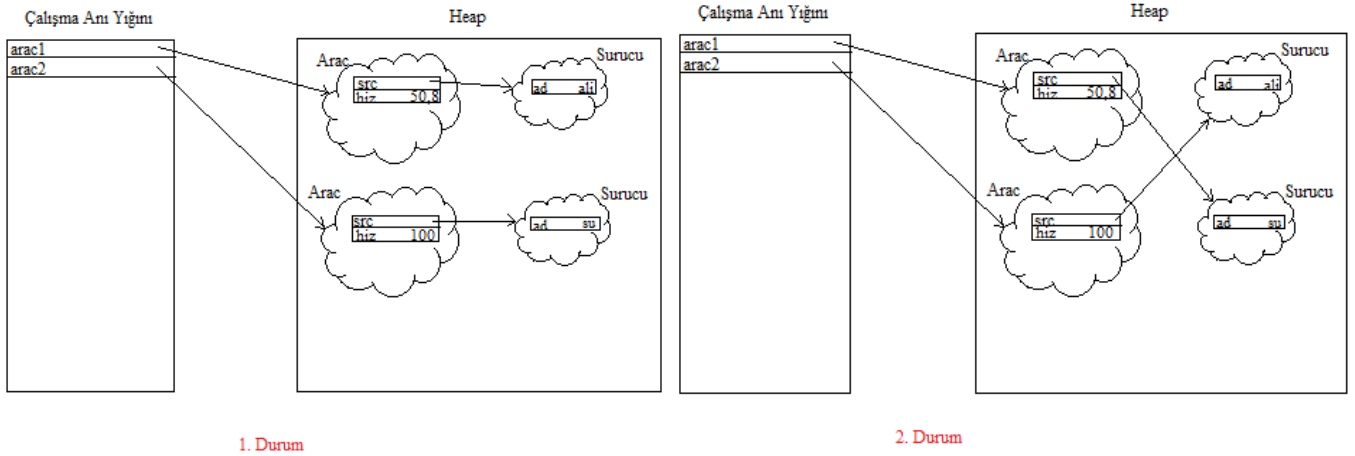
Cevap

```

void TersYaz(Dugum<Nesne> *altListe,int index=0){
    if(altListe == NULL)return;
    TersYaz(altListe->ileri,index+1);
    if(index%2==0) cout<<altListe->veri<<" ";
}
void TersYaz(){
    TersYaz(bas);
}

```

- 1.) Aşağıdaki 1. Durumu 2. Duruma getirecek kodu yazınız. Sınıfın detayları yazılmalıdır. (50 p)



1. Durum

2. Durum

Cevap:

```

#include <iostream>
using namespace std;

class Surucu{
public:
    string ad;
    Surucu(string isim) : ad(isim) {}
};

```

```

class Arac{
public:
    Surucu *src;
    float hiz;
    Arac(string isim, float hz){
        src = new Surucu(isim);
        hiz = hz;
    }
    ~Arac(){
        delete src;
    }
};

int main(){
    Arac *arac1 = new Arac("ali",50.8);
    Arac *arac2 = new Arac("su",100);

    Surucu *tmp = arac1->src;
    arac1->src = arac2->src;
    arac2->src = tmp;
    delete arac1;
    delete arac2;
    return 0;
}

```

- 2.) Tek yönlü bağlı listede listeyi tersten yazdıracak metodu yazınız. Kullanılacak bazı metotların yazılmış olduğu varsayılacaktır. (50 p)

```

void TersYaz(Dugum<Nesne> *altListe){
    if(altListe == NULL)return;
    TersYaz(altListe->ileri);
    cout<<altListe->veri<<" ";
}

void TersYaz(){
    TersYaz(bas);
}

```

- 1.) main metodu içerisine, verilen diziyi kullanarak heap'te iki araç oluşturun daha sonra bu araçların sürücülerini değiştirin. Çöp oluşmaması için kod kısmında eksik bırakılan yerleri yazınız.

```

#include <iostream>
using namespace std;

struct Surucu{
    string ad;
    Surucu(string ism) : ad(ism) { }
};

struct Arac{
    float hiz;
    Surucu *src;
    Arac(float hz, string isim){
        hiz=hiz;
        src = new Surucu(isim);
    }
    ~Arac(){

```

```

        delete src;
    }
};

int main(){
    Arac *araclar[2];
    araclar[0] = new Arac(70,"Mehmet");
    araclar[1] = new Arac(55,"Ali");
    Surucu *tmp = araclar[0]->src;
    araclar[0]->src = araclar[1]->src;
    araclar[1]->src = tmp;
    delete araclar[0];
    delete araclar[1];

    return 0;
}

```

- 2.) Bağlı liste ile gerçekleştirilmiş olan yığıtın (Stack) içindeki elemanları ekrana yazan kodu C++'ta yazınız. (50 p)

Cevap

```

void Yaz(){
    for(Dugum<Nesne> *current = topOfStack; current != NULL; current = current->ileri){
        cout<<current->eleman<<" ";
    }
}

```

- 1.) Aşağıda oluşturulan araç ve her araç içinde iki surucu kodunda, SurucuDegistir metodunun içeriğini yazınız. main metodu içerisinde bu metodu kullanınız. İki aracın sürücüleri birbiri ile değişmelidir.

```

#include <iostream>
using namespace std;

struct Surucu{
    string ad;
    Surucu(string ism) : ad(ism) { }
};

struct Arac{
    float hiz;
    Surucu *src[2];
    Arac(float hz, string isim1, string isim2){
        hiz=hz;
        src[0] = new Surucu(isim1);
        src[1] = new Surucu(isim2);
    }
    void SurucuDegistir(Surucu **srrc){
        for(int i=0;i<2;i++)
        {
            Surucu* srcTmp = src[i];
            src[i] = srrc[i];
            srrc[i] = srcTmp;
        }
    }
}

```

```

        ~Arac(){
            delete src[0];
            delete src[1];
        }
};

int main(){
    Arac *arac1,*arac2;
    arac1 = new Arac(70,"ali","mehmet");
    arac2 = new Arac(25,"murat","kadir");
    arac1->SurucuDegistir(arac2->src);
    delete arac1;
    delete arac2;
    return 0;
}

```

2.) Dizi ile gerçekleştirilmiş olan yığıtın (Stack) içindeki elemanları ekrana yazan kodu C++'ta yazınız. (50 p)

Cevap

```

void Yaz(){
    for(int i=0;i<BOYUT;i++){
        cout<<elemanlar[i]<<" ";
    }
}

```