```
// CİFT BAGLİ LİSTE UYGULAMASI

#include<iostream>
using namespace std;

struct node {
	int data;
	node *next;
	node *prev;
};

node *ekle(node *r, int x) {

	if (r == NULL)
	{
		r = new node;
		r->data = x;
		r->next = r;
		r->prev = NULL;
	}
	else
	{
		node *yrd;
		yrd = r;
		while (yrd->next != r)
			yrd = yrd->next;
		yrd->next = new node;
		yrd->next->data = x;
		yrd->next->next = r;
		yrd->next->prev = yrd;
		yrd = NULL;
		delete yrd;
	}
	return r;
}
node *basaekle(node *r, int x) {

	if (r == NULL)
	{
		r = new node;
		r->data = x;
		r->next = r;
		r->prev = NULL;
	}
	else
	{
		node *yrd;
		yrd = new node;
		yrd->next = r;
		yrd->prev = NULL;
		yrd->data = x;
		r->prev = yrd;
		yrd = yrd->next;

		while (yrd->next != r)
			yrd = yrd->next;
		//	r=r->prev;
		//	yrd->next=r;
		yrd->next = r->prev;
		r = r->prev;
		yrd = NULL;
		delete yrd;
	}
	return r;

}

void arayaekle(node *r, int x, int ara) {
```

```
// CİFT BAGLİ LİSTE UYGULAMASI


if (r == NULL)
	{
		r = new node;
		r->data = x;
		r->next = r;
		r->prev = NULL;
	}
	else
	{
		if (r->next == r)
		{
			r = ekle(r, x);
		}
		else
		{
			node *yrd;
			yrd = r;
			while (yrd->data != ara)
				yrd = yrd->next;
			node *tmp;
			tmp = new node;
			tmp->data = x;
			tmp->next = yrd->next;
			yrd->next = tmp;
			tmp->next->prev = tmp;
			tmp->prev = yrd;
			tmp = yrd = NULL;
			delete tmp, yrd;
		}
	}
}


void sil(node *r, int ara) {

	if (r == NULL)
	{
	}
	else
	{
		node *yrd;
		yrd = r;
		while (yrd->data != ara)
			yrd = yrd->next;
		yrd->prev->next = yrd->next;
		yrd->next->prev = yrd->prev;
		yrd->next = NULL;
		yrd->prev = NULL;
//data kullanilacaksa islem görsün hesapla(yrd->data);
		delete yrd;
	}
}


void yazdir(node *r) {

	if (r == NULL)
	{//arayı doldur}
	else
	{
		node *yrd;
		yrd = r;
		while (yrd->next != r) {
			cout << yrd->data << " ";
			yrd = yrd->next;
		}
		cout << yrd->data << endl;
	}
	system("pause");
}


void main() {

	node *root;
	root = NULL;
	root = ekle(root, 10);
	ekle(root, 20);
	ekle(root, 30);
	ekle(root, 40);
	ekle(root, 50);
	yazdir(root);
	root = basaekle(root, 5);
	yazdir(root);
	arayaekle(root, 35, 30);
	yazdir(root);
	sil(root, 35);
	yazdir(root);
}
```