```
//tek bagli dogrusal liste için genel uygulamalar//

#include<iostream>
using namespace std;

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

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

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

}

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

	if (r == NULL)
	{
		r = ekle(r, x);
	}
	else
	{
		node *yrd;
		yrd = new node;
		yrd->data = x;
		yrd->next = r;
		r = yrd;
		yrd = NULL;
		delete yrd;
	}
	return r;

}

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

	if (r == NULL)
		r = ekle(r, x);
	else
	{
		if (r->next == NULL)
			r = ekle(r, x);
		else
		{
			node *tmp;
			node *yrd;
			yrd = r;
			while (yrd->data != ara && yrd->next != NULL)
```

```
//tek bagli dogrusal liste için genel uygulamalar//

                        yrd = yrd->next;
                    if (yrd->data == ara)
                    {
                        tmp = yrd->next;
                        yrd->next = new node;
                        yrd->next->data = x;
                        yrd->next->next = tmp;
                        tmp = yrd = NULL;
                        delete yrd, tmp;
                    }
                    else
                    {
                        cout << "eleman bulunamadi ama sona ekledim" << endl;
                        r = ekle(r, x);
                    }
                }
            }
}

void bastir(node *r) {

        if (r == NULL)
            cout << "eleman yok";
        else
        {
            while (r != NULL)
            {
                cout << r->data << " ";
                r = r->next;
            }
        }

        cout << endl;

        system("pause");
}

void main() {

        node *root;
        root = NULL;
        root = ekle(root, 8);
        ekle(root, 18);
        ekle(root, 38);

        ekle(root, 48);
        ekle(root, 58);
        ekle(root, 68);
        ekle(root, 78);

        bastir(root);
        root = basaekle(root, 7);
        bastir(root);
        arayaekle(root, 37, 40);
        bastir(root);
        arayaekle(root, 68, 54);
        bastir(root);
        ekle(root, 118);
        ekle(root, 128);
        ekle(root, 138);
        ekle(root, 178);
        bastir(root);

}
```