

```

#include <iostream>
using namespace std;
class lista;
class nodo
{
friend class lista;
private:
int info;
nodo *sig;
public:
nodo(int,nodo *);
};
nodo::nodo(int x,nodo *p=0):info(x),sig(p){ }
class lista
{
private:
nodo *p;
public:
lista(); //constructor sin argumentos; agregar el constructor copia
~lista(); //destructor
void altafin(int); // alta al final
void altaprin(int); //alta al principio
void baja(int); //baja de un elemento determinado
bool esta(int); // retorna un bool indicando si el entero pasado por argumento está en la lista
void emite(); //emite la lista
};
void lista::altafin(int x)
{
nodo *aux=p;
while ((aux)&&(aux->sig))
aux=aux->sig;
if (aux)
{
aux->sig=new nodo(x,aux->sig);
}
else p=new nodo(x,p);
}

void lista::altaprin(int x)
{
p=new nodo(x,p);
}

void lista::baja(int x)
{
if (p)
{
if((p->info!=x) && (p->sig))
{
nodo *aux1 =p, *aux2 =p->sig;
while ((aux2->info!=x) && (aux2->sig))
{
aux1=aux2 ;
aux2 =aux2->sig;
}
if (aux2->info==x)

```

```

{
aux1->sig=aux2->sig;
delete aux2;
}
}
else if(p->info==x)
{
nodo *aux =p;
p=p->sig;
delete aux;
}
}
}
lista::~lista()
{
if (p)
{
nodo *aux=p;
while (aux)
{
p=p->sig;
delete aux;
aux=p;
}
}
}
bool lista::esta(int x)
{
nodo *aux=p;
while(aux)
{
if (aux->info == x) return true;
else aux=aux->sig;
}
return false;
}
void lista::emite()
{
nodo *aux=p;
while (aux)
{
cout<<aux->info;
aux=aux->sig;
}
}
lista::lista():p(0){ }
void main()
{
lista l;
cout<<"\n alta al final de 1\n";
l.altafin(1);
l.emite();
cout<<"\n alta al final de 2\n";
l.altafin(2);
l.emite();
cout<<"\n alta al final de 3\n";

```

```
l.altafin(3);
l.emite();
cout<<"\n alta al final de 4\n";
l.altafin(4);
l.emite();
cout<<"\n alta al principio de 5\n";
l.altaprin(5);
l.emite();
cout<<"\n baja 4\n";
l.baja(4);
l.emite();
cout<<"\n baja 1\n";
l.baja(1);
l.emite();

}
```