

```

void righth_baja(nodo_grafo*&grafo){//borro el conjunto derecho(lo que sale de un nodo)
nodo_arco*actual=grafo->lista_arc;nodo_arco *anterior=NULL;
while(grafo->lista_arc!=NULL){
while(actual!=NULL){
anterior=actual;
actual=actual->link;
delete anterior;
}
grafo->lista_arc=NULL;
cout<<"borre un arco"<<endl;

}

}

```

```

bool borrar_arco_primero(nodo_arco*&lista_arc,nodo_grafo*dir){//borro la primera iteracion de un arco con
la clave pasada como parametro
nodo_arco *actual=lista_arc; nodo_arco *anterior=NULL;
while(actual!=NULL){
if(actual->destino==dir){
if(anterior==NULL){
lista_arc=actual->link;
cout<<"borre el arco " <<actual->id_arco;
delete actual;
return true;
}else{
anterior->link=actual->link;
cout<<"borre";
delete actual;
return true;
}
}
anterior=actual;
actual=actual->link;
}
return false;

}

```

```

void left_baja(nodo_grafo*grafo,nodo_grafo* dir_nodo){borro el conjunto izquierdo de un nodo(lo que llega)
nodo_grafo *actual=grafo;
while(actual!=NULL){
while(borrar_arco_primero(actual->lista_arc,dir_nodo));
actual=actual->link;
}
}

```