

```

void eliminar()
{
    char nom[20], band='F'; int sgte, max_izq,aux,aux2;
    if((fd=fopen("personass.txt","r+t")) != NULL){
        fread(&e,le,1,fd);
        cout<<"nombre:"; fflush(stdin); gets(nom);
        //algoritmo buscar al registro que apunta al nuevo
        sgte= e.RZ;

        while(sgte != -1){
            pos=(sgte-1)*lr + le;
            fseek(fd,pos,0);
            fread(&s,lr,1,fd);
            if(strcmp(nom,s.nombre)>0){
                aux=sgte;
                sgte=s.PD; continue;
            }
            aux2=max_izq;

            max_izq = reg_max_izq.PD;
        }

        //caso B
        char nomP[20];
        char celP[10];
        cout<<"Máximo izquierdo:"<<reg_max_izq.nombre<<endl;
        strcpy(nomP,reg_max_izq.nombre);
        strcpy(celP,reg_max_izq.celular);
        cout<<"nombre "<<nomP;
        cout<<"cel "<<celP;
        strcpy(reg_max_izq.nombre,s.nombre);
    }
    band = 'V';
    //se encontró la clave, ahora busquemos su max_izquierdo

    if(s.PI != -1){
        //existe un maximo izquierdo
        max_izq = s.PI;
        aux2=s.NR;
        while(max_izq != - 1){
            pos = (max_izq -1)*lr + le;
            fseek(fd,pos,0);
            fread(&reg_max_izq,lr,1,fd);
        }
    }
}

```

```

        if(reg_max_izq.PD!=-1){
            aux2=max_izq;
        }

        max_izq = reg_max_izq.PD;
    }
    I
    ///caso B
    char nomP[20];
    char celP[10];
    cout<<"Máximo izquierdo:"<<reg_max_izq.nombre<<endl;
    strcpy(nomP,reg_max_izq.nombre);
    strcpy(celP,reg_max_izq.celular);
    cout<<"nombre "<<nomP;
    cout<<"cel "<<celP;
    strcpy(reg_max_izq.nombre,s.nombre);
    strcpy(reg_max_izq.celular,s.celular);
    cout<<"nombre "<<nomP;
    cout<<"cel "<<celP<<endl;
    strcpy(s.nombre,nomP);
    strcpy(s.celular,celP);
    cout<<"nombre "<<s.nombre;
    cout<<"cel "<<s.celular<<endl;
    reg_max_izq.ARE=e.UR;
    e.UR=reg_max_izq.NR;
    pos=(reg_max_izq.NR-1)*lr + le;

    fseek(fd,pos,0);
    fwrite(&reg_max_izq,lr,1,fd);
    I

    pos = (aux2 -1)*lr + le;
    fseek(fd,pos,0);
    fread(&r,lr,1,fd);
    r.PD=reg_max_izq.PI;

    fseek(fd,pos,0);

```

```

        fwrite(&r,lr,1,fd);
    }
    else{
        ///caso A
        cout<<"no tiene maximo izquierdo"<<endl;
        s.ARE=e.UR;
        e.UR=s.NR;

        I
        pos=(aux-1)*lr + le;
        fseek(fd,pos,0);
        fread(&r,lr,1,fd);
        r.PI=s.PD;
        cout<<"n: "<<r.nombre;
        cout<<"nr: "<<r.NR;
    }

```

```

    fseek(fd,pos,0);
    fwrite(&r,lr,1,fd);
    pos=(s.NR-1)*lr + le;
    fseek(fd,pos,0);
    fwrite(&s,lr,1,fd);

```

```

    fseek(fd,0,I,0);
    fwrite(&e,lr,1,fd);

```

```

//fin de busqueda de maximo izquierdo
    //fin de busqueda de maximo izquierdo
    break;
}

```

```

//fin de algoritmo buscar
if(band=='F'){cout<<"nombre no existe"<<endl;}
fclose(fd);

```

```

}

```

```
else  
|   cout<<"no se pudo abrir el archivo"<<endl;  
  
}
```