Int main()

lVues ← NULL
Abrir(Aerops, Vues, Conslts)
ProcAeropuertos (Aerops, vrAerop)
ProcVuelos(Vues, lVues)
ConsultasVuelos(Conslts, Vues, Aerops, lVues, vrAerop)
ListVueAeropSld(Aerops, Vues, vrAerop, lVues)
Cerrar(Aerops, Vues, Conslts)
retornar 0

R

long GetTime(**int** S:hh, **int** S:mm, **int** S:ss)

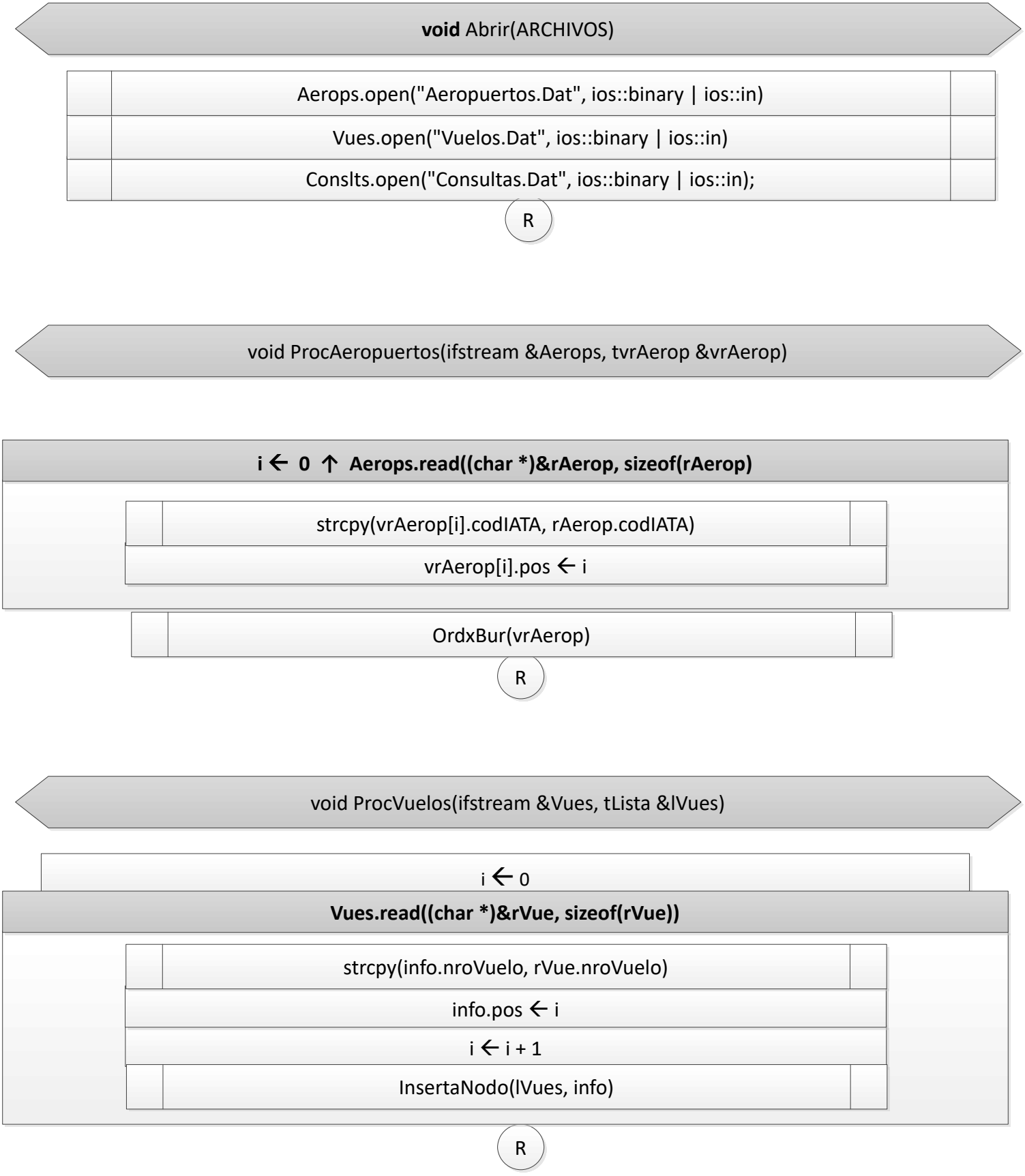
time(&rawtime)
timeinfo ← localtime(&rawtime)
hh ← timeinfo.tm_hour
mm ← timeinfo.tm_min
ss ← timeinfo.tm_sec
retornar timeinfo.tm_hour * 10000 + timeinfo.tm_min * 100 + timeinfo.tm_sec

R

long GetDate(**int** S:year, **int** S:mes, **int** S:dia, **int** S:ds)

time(&rawtime)
timeinfo ← localtime(&rawtime)
year ← 1900 + timeinfo.tm_year
mes ← 1 + timeinfo.tm_mon
dia ← timeinfo.tm_mday
ds ← 1 + timeinfo.tm_wday
retornar (1900 + timeinfo.tm_year) * 10000 + (1 + timeinfo.tm_mon) * 100 + timeinfo.tm_mday

R


void Abrir(ARCHIVOS)
Aerops.open("Aeropuertos.Dat", ios::binary | ios::in)
Vues.open("Vuelos.Dat", ios::binary | ios::in)
Conslts.open("Consultas.Dat", ios::binary | ios::in);
R
void ProcAeropuertos(ifstream &Aerops, tvrAerop &vrAerop)
i ← 0 ↑ Aerops.read((char *)&rAerop, sizeof(rAerop)
strcpy(vrAerop[i].codIATA, rAerop.codIATA)
vrAerop[i].pos ← i
OrdxBur(vrAerop)
R
void ProcVuelos(ifstream &Vues, tLista &lVues)
i ← 0
Vues.read((char *)&rVue, sizeof(rVue))
strcpy(info.nroVuelo, rVue.nroVuelo)
info.pos ← i
i ← i + 1
InsertaNodo(lVues, info)
R

void ConsultasVuelos(ifstream &Conslts, ifstream &Vues, ifstream &Aerops,
tLista &lVues, tvrAerop &vrAerop)

freopen("Listado Consulta Vuelos.Txt", "w", stdout);

aux ← lVues

GetTime(hh, mm, ss)

GetDate(anio, mes, dia, ds)

"Consultas de vuelos del ", dia, " de ", meses[mes], " de ", anio
"NroVuelo Ciudad de origen Nom.Aerop.Orig. Empresa Marca "
" Ciudad Destino Nom.Aerop.Dest. Estado dia hhAct hhSa "
"t.V. hhLl"

Conslts.read(nroVuelo, sizeof(str9))

strcmp(aux->info.nroVuelo, nroVuelo) > 0

aux ← lVues

strcmp(aux->info.nroVuelo, nroVuelo) != 0

aux ← aux->sgte

Vues.clear()

Vues.seekg(aux->info.pos * sizeof(sVue))

Vues.read((char *)&rVue, sizeof(sVue))

Aerops.clear()

Aerops.seekg(vrAerop[BusBinVec(vrAerop, nroVuelo)].pos * sizeof(sAerop))

Aerops.read((char *)&origen, sizeof(sAerop))

Aerops.clear()

Aerops.seekg(vrAerop[BusBinVec(vrAerop, nroVuelo + 6)].pos * sizeof(sAerop));

Aerops.read((char *)&destino, sizeof(sAerop))

diaSa ← rVue.fechaSale % 100

mesSa ← rVue.fechaSale / 100 % 100

anioSa ← rVue.fechaSale / 10000

FormatoHoraMin(rVue.horaSale, hhSa, mmSa)

HoraLlega(rVue.distKm, rVue.velCrucero, hhSa, mmSa, hhVi, mmVi, hhLl, mmLl)

VerifEstado(estado, hhSa, mmSa, hhLl, diaSa, mesSa, anioSa)

origen.ciudad[16] ← '\0'

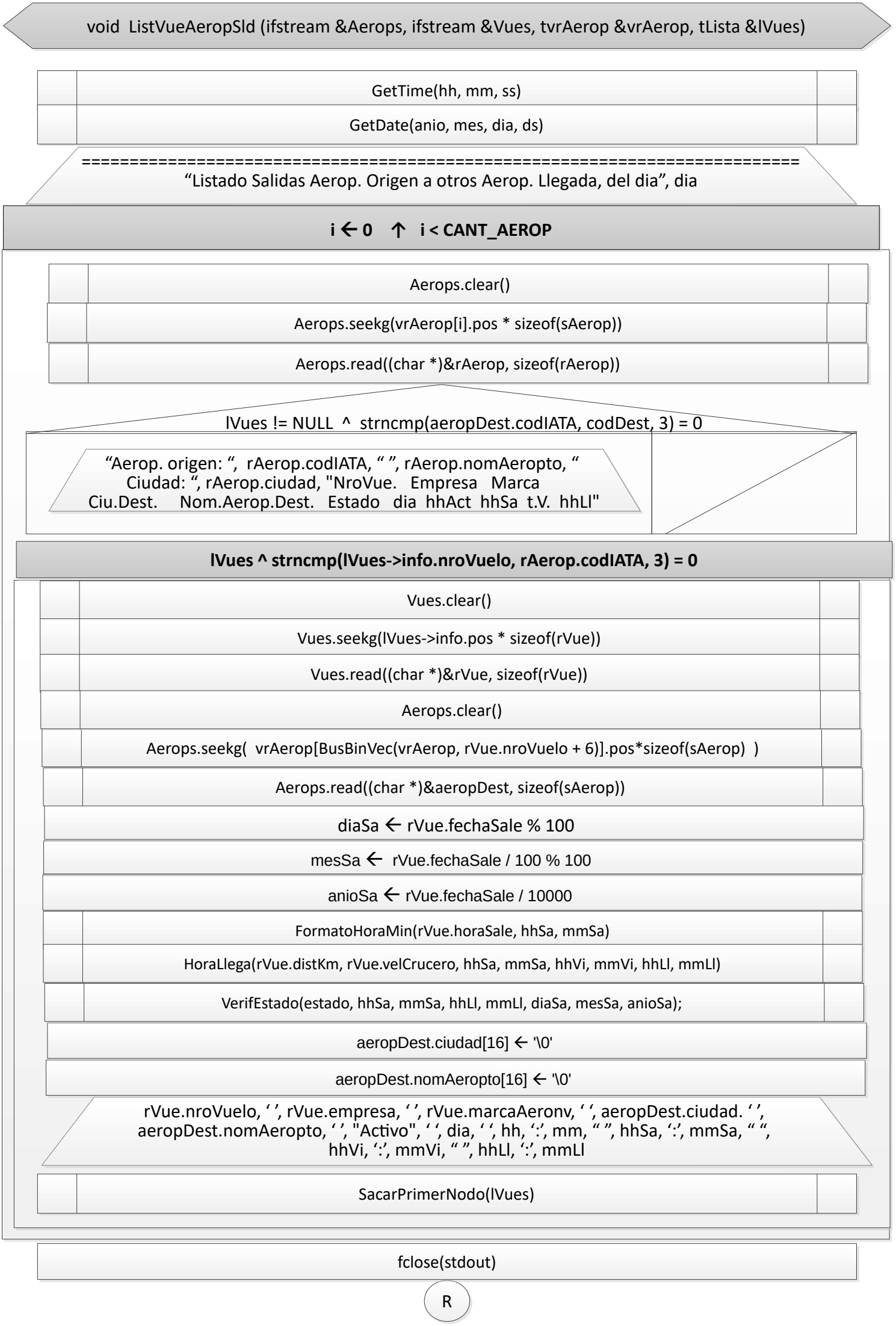
origen.nomAeropto[16] ← '\0'

destino.ciudad[16] ← '\0'

destino.nomAeropto[16] ← '\0'

rVue.nroVuelo, ' ', origen.ciudad,' ',origen.nomAeropto,' ',rVue.empresa,' ',rVue.marcaAeronv,
' ',destino.ciudad,' ',destino.nomAeropto,' ',destino.provin,' ',diaSa,' ',hh,':',mm,' ',hhSa,':',
MmSa,' ',hhVi,':',mmVi,' ',hhLl,':',mmLl

R


void ListVueAeropSld (ifstream &Aerops, ifstream &Vues, tvrAerop &vrAerop, tLista &lVues)
GetTime(hh, mm, ss)
GetDate(anio, mes, dia, ds)
==========
"Listado Salidas Aerop. Origen a otros Aerop. Llegada, del dia", dia
i ← 0 ↑ i < CANT_AEROP
Aerops.clear()
Aerops.seekg(vrAerop[i].pos * sizeof(sAerop))
Aerops.read((char *)&rAerop, sizeof(rAerop))
lVues != NULL ^ strncmp(aeropDest.codIATA, codDest, 3) = 0
"Aerop. origen: ", rAerop.codIATA, " ", rAerop.nomAeropto, "
Ciudad: ", rAerop.ciudad, "NroVue. Empresa Marca
Ciu.Dest. Nom.Aerop.Dest. Estado dia hhAct hhSa t.V. hhLl"
lVues ^ strncmp(lVues->info.nroVuelo, rAerop.codIATA, 3) = 0
Vues.clear()
Vues.seekg(lVues->info.pos * sizeof(rVue))
Vues.read((char *)&rVue, sizeof(rVue))
Aerops.clear()
Aerops.seekg(vrAerop[BusBinVec(vrAerop, rVue.nroVuelo + 6)].pos*sizeof(sAerop))
Aerops.read((char *)&aeropDest, sizeof(sAerop))
diaSa ← rVue.fechaSale % 100
mesSa ← rVue.fechaSale / 100 % 100
anioSa ← rVue.fechaSale / 10000
FormatoHoraMin(rVue.horaSale, hhSa, mmSa)
HoraLlega(rVue.distKm, rVue.velCrucero, hhSa, mmSa, hhVi, mmVi, hhLl, mmLl)
VerifEstado(estado, hhSa, mmSa, hhLl, mmLl, diaSa, mesSa, anioSa);
aeropDest.ciudad[16] ← '\0'
aeropDest.nomAeropto[16] ← '\0'
rVue.nroVuelo, ' ', rVue.empresa, ' ', rVue.marcaAeronv, ' ', aeropDest.ciudad. ' ',
aeropDest.nomAeropto, ' ', "Activo", ' ', dia, ' ', hh, ':', mm, " ", hhSa, ':', mmSa, " ",
hhVi, ':', mmVi, " ", hhLl, ':', mmLl
SacarPrimerNodo(lVues)
fclose(stdout)
R

void Cerrar(ARCHIVOS)

Aerops.close()

Vues.close()

Conslts.close()

R

void OrdxBur(tvrAerop S:vrAerop)

k ← 0

hayCambios ← false

k ← k+1

i ← 0 ↑ CANT_AEROP - k

strcmp(vrAerop[i].codIATA, vrAerop[i + 1].codIATA) > 0

IntCmb(vrAerop[i], vrAerop[i + 1])

hayCambios ← true

hayCambios

R

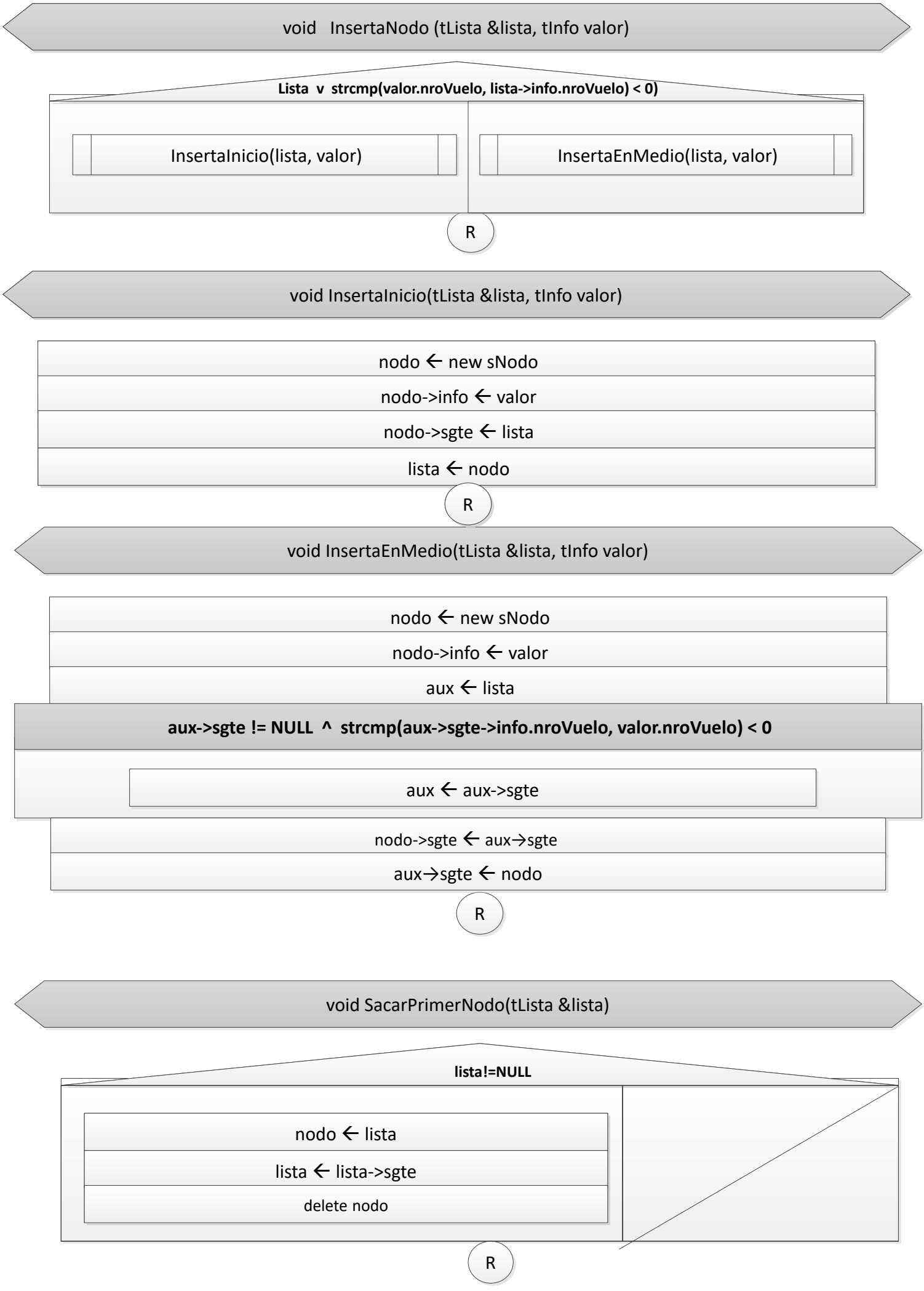
void IntCmb(sTblAerop S: sElem1, sTblAerop S: sElem2)

auxiliar ← sElem1

sElem1 ← sElem2

sElem2 ← auxiliar

R


void InsertaNodo (tLista &lista, tInfo valor)
Lista v strcmp(valor.nroVuelo, lista->info.nroVuelo) < 0)
InsertaInicio(lista, valor)
InsertaEnMedio(lista, valor)
R
void InsertaInicio(tLista &lista, tInfo valor)
nodo ← new sNodo
nodo->info ← valor
nodo->sgte ← lista
lista ← nodo
R
void InsertaEnMedio(tLista &lista, tInfo valor)
nodo ← new sNodo
nodo->info ← valor
aux ← lista
aux->sgte != NULL ^ strcmp(aux->sgte->info.nroVuelo, valor.nroVuelo) < 0
aux ← aux->sgte
nodo->sgte ← aux→sgte
aux→sgte ← nodo
R
void SacarPrimerNodo(tLista &lista)
lista!=NULL
nodo ← lista
lista ← lista->sgte
delete nodo
R

int BusBinVec(tvrAerop &vrAerop, str3 codIATA)

li ← 0

ls ← CANT_AEROP - 1

li <= ls

pm ← (li + ls) / 2

cpm ← strcmp(codIATA, vrAerop[pm].codIATA, 3)

cmp = 0

retornar pm

cmp < 0

ls ← pm - 1

li ← pm + 1

retornar -1

R

void FormatoHoraMin(short hora, short &hh, short &mm)

hh ← hora/100

mm ← hora – hh *100

R

void HoraLlega(short distKm, short velCrucero, short hhSa, short mmSa,
short &hhVi, short &mmVi, short &hhLl, short &mmLl)

tV ← (float)distKm / (float)velCrucero

hhVi ← tV

mmVi ← (tV - hhVi) * 60

mmLl ← (mmSa + mmVi) % 60

hhLl ← (hhSa + hhVi + (mmSa + mmVi) / 60) % 24

R

string Replicate(**char** car, **ushort** n)

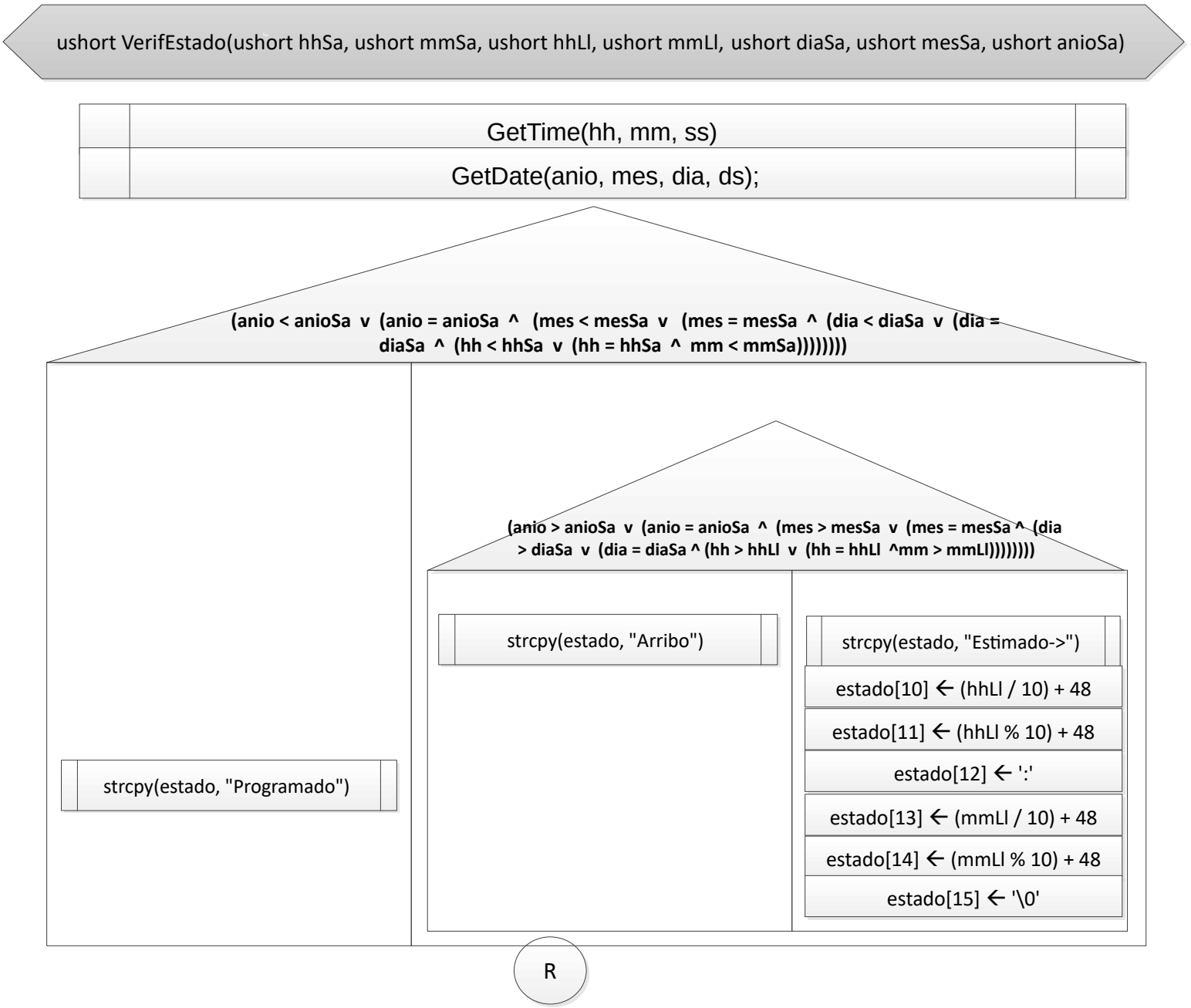
resultado ← “”

i ← 0 ↑ i < n

resultado ← resultado + car

retornar resultado

R


ushort VerifEstado(ushort hhSa, ushort mmSa, ushort hhLl, ushort mmLl, ushort diaSa, ushort mesSa, ushort anioSa)
GetTime(hh, mm, ss)
GetDate(anio, mes, dia, ds);
(anio < anioSa v (anio = anioSa ^ (mes < mesSa v (mes = mesSa ^ (dia < diaSa v (dia = diaSa ^ (hh < hhSa v (hh = hhSa ^ mm < mmSa)))))))
strcpy(estado, "Programado")
(anio > anioSa v (anio = anioSa ^ (mes > mesSa v (mes = mesSa ^ (dia > diaSa v (dia = diaSa ^ (hh > hhLl v (hh = hhLl ^mm > mmLl)))))))
strcpy(estado, "Arribo")
strcpy(estado, "Estimado->")
estado[10] ← (hhLl / 10) + 48
estado[11] ← (hhLl % 10) + 48
estado[12] ← ':'
estado[13] ← (mmLl / 10) + 48
estado[14] ← (mmLl % 10) + 48
estado[15] ← '\0'
R