

Int main()

	Abrir(Art, IndDesc, Rub, ListCmpr)	
	VolcarArchivos(Art, IndDesc, Rub, ListCmpr, vsArt, vsIndDesc, vsRub, vsListCmpr, cantArt, cantCmpr)	
	ProcCompras(Art, vsArt, vsIndDesc, vsListCmpr, cantArt, cantCmpr)	
	EmitirTicket(vsArt, vsIndDesc, vsListCmpr, cantArt, cantCmpr)	
	EmitirArt_x_Rubro(vsArt, vsRub, cantArt)	
	Cerrar(Art, IndDesc, Rub, ListCmpr)	
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

bool LeerArticulo(ifstream &Art, tsArt &sArt)

Art >> sArt.codVen >> sArt.codRub

Art.ignore()

Art.get(sArt.descArt, 31)

Art >> sArt.stock >> sArt.preUni

Art.ignore()

Art.get(sArt.medida, 31)

i ← 0 ↑ i < 14

Art >> sArt.ofertas[i]

Art.ignore(2, '\n')

retornar Art.good()

R

bool LeerIndDescrip(ifstream &IndDesc, tsIndDesc &sIndDesc)

IndDesc.get(sIndDesc.descArt, 31)

IndDesc >> sIndDesc.posArt >> sIndDesc.estado

IndDesc.ignore(2, '\n')

strlwr(sIndDesc.descArt)

retornar IndDesc.good()

R

bool LeerRubro(ifstream &Rub, tsRub &sRub)

Rub >> sRub.codRub

Rub.ignore()

Rub.get(sRub.descRub, 21)

Rub.ignore(2, '\n')

retornar Rub.good()

R

bool LeerCompra(ifstream &ListCmpr, tsCompra &sCompra)

ListCmpr.get(sCompra.descArt, 31)

ListCmpr >> sCompra.cantReq

ListCmpr.ignore(2, '\n')

retornar ListCmpr.good()

R

void PieTicket(**float** impTot, **float** impTotDesto, **float** impTotConDesto)

PagoUsuario ← impTotConDesto

vuelto ← pagoUsuario - impTotConDesto

```

    "SubTot. sin descuentos....: $ ", impTot
    "Descuentos por promociones: $ ", -impTotDesto
    "=====
    "T O T A L $ ", impTotConDesto
    "=====
    "Su pago con Mercado Pago: $ ", pagoUsuario
    "Su vuelto: $ ", vuelto
    "   G R A C I A S   P O R   S U   C O M P R A
    "   Para consultas, sugerencias o reclamos
    "   comunicarse al correo infoKotto.com.ar"

```

R

void CabeceraTicket(**int** &ds)

GetTime(hh, mm, ss)

GetDate(anio, mes, dia, ds)

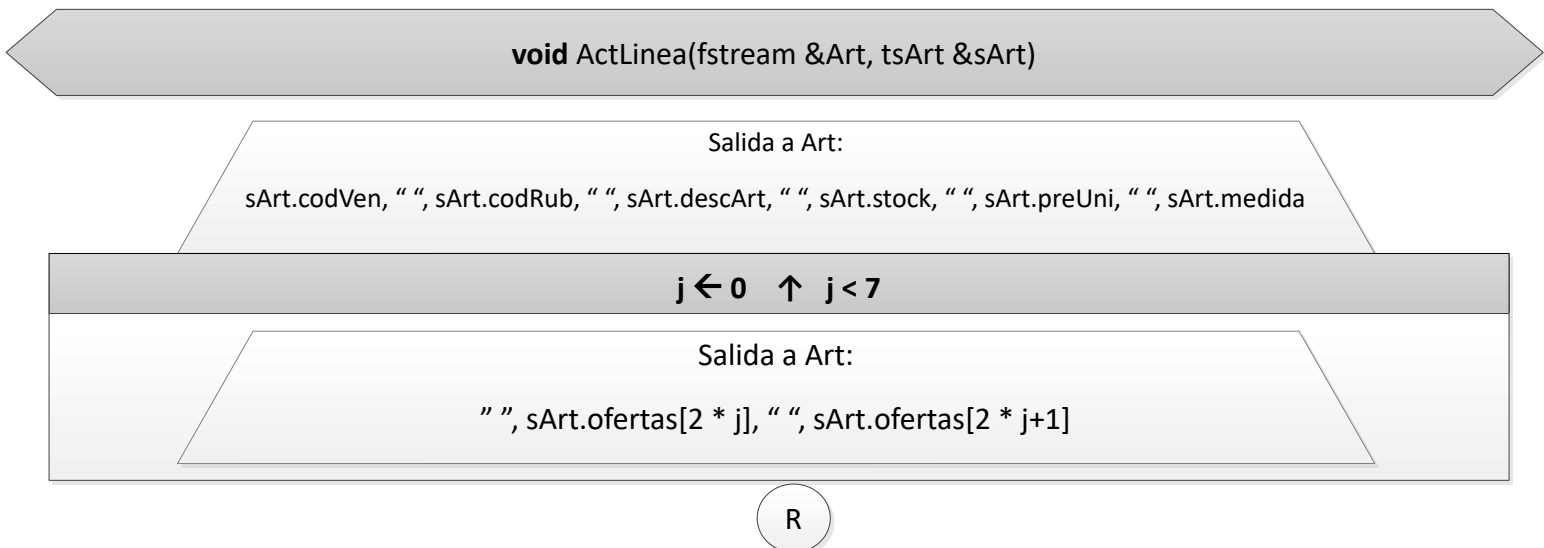
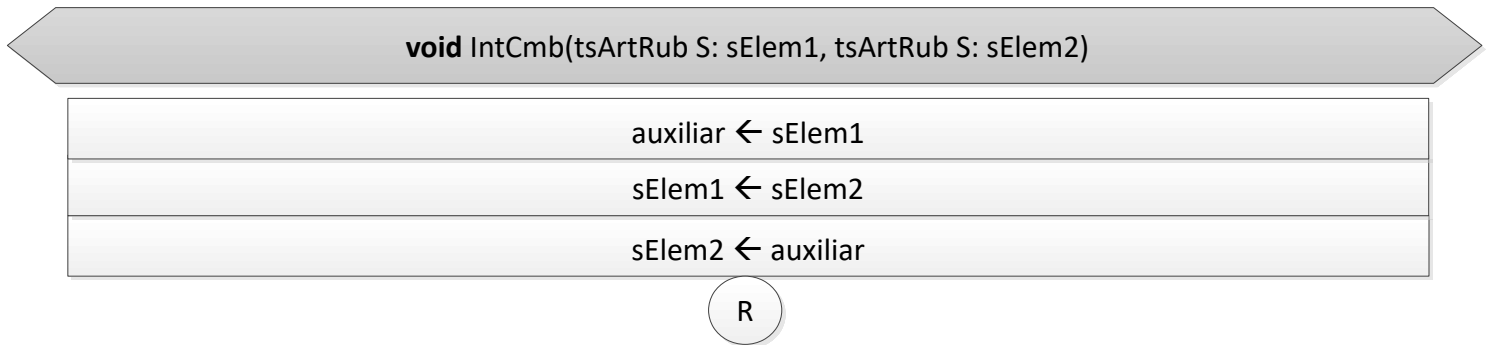
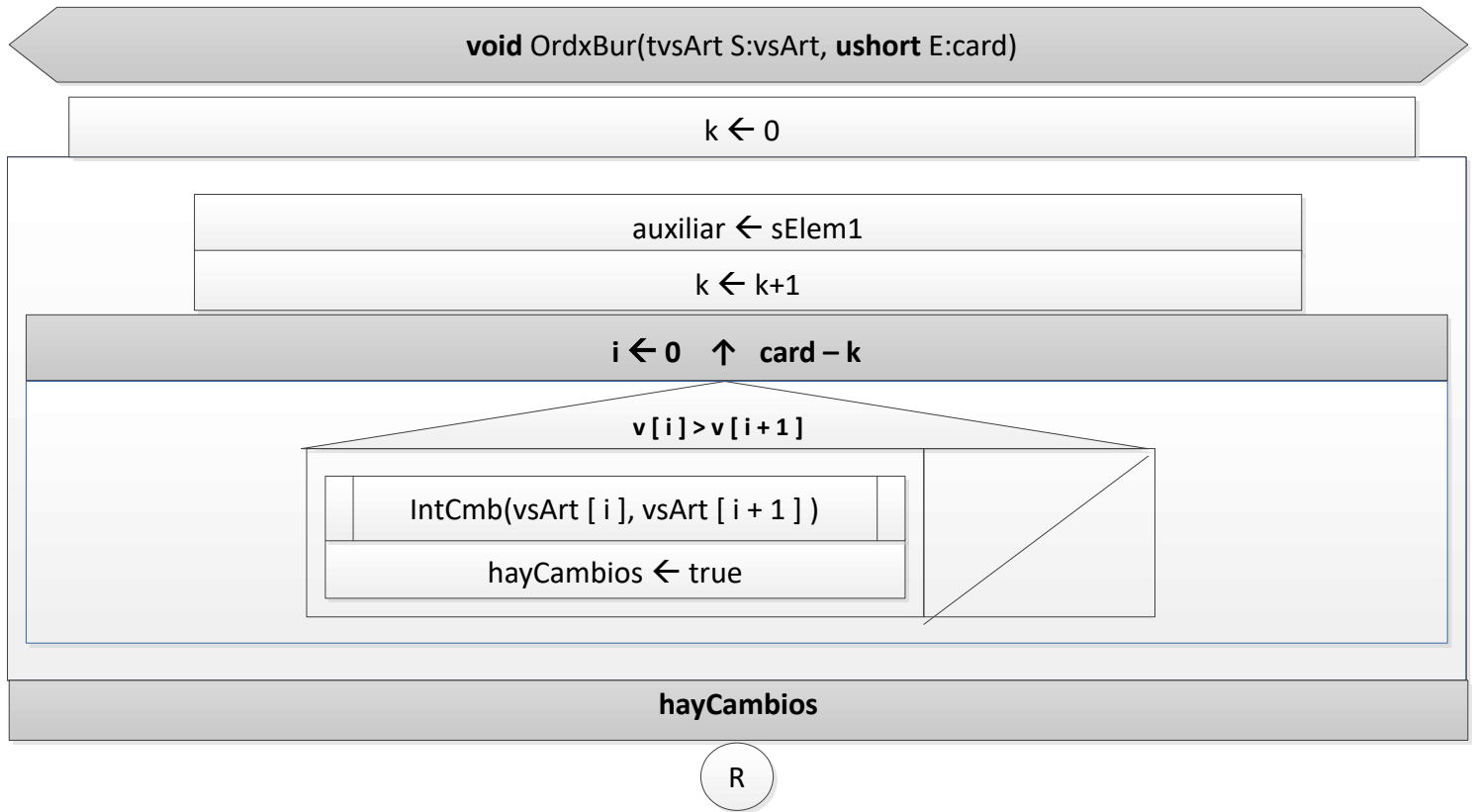
*diasSemana[] ← {"Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado"}

```

    "K O T T O"
    "Yo te reconozco"
    "SUC 170"
    "XXXXXX...X 9999"
    "XX...X"
    "C.U.I.T. 99-99999999-9"
    "Fecha: ", diasSemana[ds - 1], " ", dia, "/", mes, "/", anio,
    "Hora: ", hh, ":", mm, ":", ss,
    "Nro. Ticket: 9999-99999999"
    "Nro. Caja: 9999"
    "-----"
    "F A C T U R A - B"
    "ORIGINAL"
    "-----"

```

R



int BusBinVec(tvslndDesc S:vsIndDesc, str30 E:descArt, **ushort** E:ult)

$li \leftarrow 0$

$ls \leftarrow ult$

strlwr(descArt)

li <= ls

$pm \leftarrow (li + ls) / 2$

$cpm \leftarrow \text{strcmp}(\text{descArt}, \text{vsIndDesc}[pm].\text{descArt})$

cmp = 0

retornar pm

cmp < 0

$ls \leftarrow pm - 1$

$li \leftarrow pm + 1$

retornar -1

R

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

resultado \leftarrow ""

i \leftarrow 0 \uparrow i < n

resultado \leftarrow resultado + car

retornar resultado

R

void Abrir(ARCHIVOS)

Art.open("Articulos.txt")

IndDesc.open("IndDescripArt.txt")

Rub.open("Rubros.txt")

ListCmpr.open("ListaCompras.txt")

R

void VolcarArchivos(ARCHIVOS, REGISTROS, **ushort** S:cantArt, **ushort** S:cantCmpr)

cantArt \leftarrow 0

cantCmpr \leftarrow 0

LeerArticulo(Art, sArt) ^ cantArt < MAX_ART

vsArtRub[cantArt].codRub \leftarrow sArt.codRub

vsArtRub[cantArt].posArt \leftarrow cantArt

cantArt \leftarrow cantArt + 1

i \leftarrow 0 \uparrow LeerDescripcion(IndDesc, sIndDesc) ^ i < cantArt

vsIndDesc[i] \leftarrow sIndDesc

i \leftarrow 0 \uparrow LeerRubro(Rub, sRub) ^ i < CANT_RUB

vsRub[i] \leftarrow sRub

LeerCompra(ListCmpr, sCompra) ^ cantCmpr < MAX_COMPRAS

vsListCmpr[cantCmpr] \leftarrow sCompra

cantCmpr \leftarrow cantCmpr + 1

OrdxBur(vsArtRub, cantArt)

R

void ProcCompras(fstream &Art, REG_COMPRAS, **ushort** E:cantArt, **ushort** E:cantCmpr)

$i \leftarrow 0$ \uparrow $i < \text{cantCmpr}$

strcpy(descBuscada, vsListCmpr [i].descArt)

pos \leftarrow BusBinVec(vsIndDesc, descBuscada, cantArt - 1)

pos $\sim=$ -1 \wedge vsIndDesc[pos].estado

posArt \leftarrow vsIndDesc[pos].posArt

Art.clear()

Art.seekp(105 * posArt)

LeerArticulo(Art, sArt)

sArt.stock \geq vsListCmpr[i].cantReq

sArt.stock \leftarrow sArt.stock -
vsListCmpr[i].cantReq

vsListCmpr[i].cantReq \leftarrow sArt.stock

sArt.stock \leftarrow 0

vsListCmpr[i].cantReq \leftarrow 0

Art.clear()

Art.seekp(105 * posArt)

ActLinea(Art, vsArt[posArt])

R

void EmitirTicket(fstream &Art, tvsIndDesc &vsIndDesc, tvsListCmpr &vsListCmpr, ushort cantArt, ushort cantCmpr)

impTot \leftarrow 0.0f

impTotDesto \leftarrow 0.0f

freopen("ticket.txt", "w", stdout)

CabeceraTicket(ds)

i \leftarrow 0 \uparrow i < cantCmpr

vsListCmpr[i].cantReq > 0

pos \leftarrow BusBinVec(vsIndDesc, vsListCmpr[i].descArt, cantArt - 1);

pos > -1

Art.clear()

Art. seekp(105 * vsIndDesc[pos].posArt);

LeerArticulo(Art, sArt);

cant \leftarrow vsListCmpr[i].cantReq

precio \leftarrow sArt.preUni

subtotal \leftarrow cant * precio

tipo \leftarrow sArt.ofertas[(ds - 1) * 2]

porc \leftarrow sArt.ofertas[(ds - 1) * 2 + 1]

descuento \leftarrow 0.0f

tipo >= 2 ^ tipo <= 7

descuento \leftarrow subtotal * porc / 100.0f

tipo

case 2	case 3	case 4	case 5	case 6	case 7	default
strcpy(strDesc, "Promo")	strcpy(strDesc, "Marca")	strcpy(strDesc, "Jub.")	strcpy(strDesc, "Comunid.")	strcpy(strDesc, "MercPago")	strcpy(strDesc, "ANSES")	strcpy(strDesc, "SinPromo")

" x \$ ", precio
sArt.descArt, " ", sArt.medida
Sart.codVen, "\$ ", subtotal

descuento > 0.0f

StrDesc, porc, "\$ ", -descuento

impTot \leftarrow impTot + subtotal

impTotDesto \leftarrow impTotDesto + descuento

impTotConDesto \leftarrow impTot - impTotDesto

PieTicket(impTot, impTotDesto, impTotConDesto)

void EmitirArt_x_Rubro(fstream &Art, tvsArtRub &vsArtRub, tvsRub &vsRub, ushort cantArt)

$\text{codRubro} \leftarrow 200$

$\text{posRubro} \leftarrow -1$

“-----”
“Listado de Articulos ordenados porCodigo de Rubro”
“-----”

$i \leftarrow 0 \quad \uparrow \quad i < \text{cantArt}$

Art.clear()

Art.seekp(105 * vsArtRub[i].posArt)

LeerArticulo(Art, sArt)

$i \sim= 0$

“

“

$\text{codRubro} \sim= \text{sArt.codRub}$

$\text{codRubro} \leftarrow \text{sArt.codRub}$

$\text{posRubro} \leftarrow \text{posRubro} + 1$

$\text{PosRubro} < 15 \wedge \text{codRubro} \sim= \text{vsRub}[\text{posRubro}].\text{codRub}$

“Cod. Rubro: “, codRubro, “ “, vsRub [posRubro].descRub

“Cod.Art. Descripción”, “Stk. Prec.Uni. Uni.Medida TD % TD % TD % TD % TD % TD %”

“-----”

Sart.codVen, “ “, sArt.descArt, “ “, sArt.stock, “ “, sArt.preUni, “ “, sArt.medida

$j \leftarrow 0 \quad \uparrow \quad j < 7$

“ “, sArt.ofertas[2 * j], “ “, sArt.ofertas[2 * j + 1]

fclose(stdout)

R

void Cerrar(ARCHIVOS)

Art.close()

IndDesc.close()

Rub.close()

ListCmpr.close()

R