//TEMA 2

//ejercicio 1

function cantidadRepartidoresAEntregarPaquete(){

/\*

PROPÓSITO: Describe la cantidad de repartidores que deben entregar sus paquete, estos paquetes deben estar en el mapa.

PRECONDICIONES:

\*Ninguna

TIPO: Número

\*/

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadRepartidores := unoSi\_ceroSino(hayRepartidorConPaqueteParaEntregarEnMapa())

while(haySiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadRepartidores := cantidadRepartidores + unoSi\_ceroSino(hayRepartidorConPaqueteParaEntregarEnMapa())

}

return(cantidadRepartidores)

}

function hayRepartidorConPaqueteParaEntregarEnMapa(){

/\*

PROPÓSITO: Describe si hay un repartidor en la ubicación actual que tiene paquete en el mapa sin entregar.

PRECONDICIONES:

\*Ninguna

TIPO: Booleano

\*/

return(hayRepartidorAcá() && hayPaqueteDel\_EnElMapa(idRepartidorAcá()))

}

function hayPaqueteDel\_EnElMapa(idRepartidor){

/\*

PROPÓSITO: Indica si hay al menos un paquete con código \*\*idRepartidor\*\* en el mapa.

PARÁMETROS:

\*idRepartidor: Número- el id del código del paquete a buscar

PRECONDICIONES:

\*Ninguna

TIPO: Booleano

\*/

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

while(haySiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte) && not hayPaqueteConIdCódigo\_(idRepartidor)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

}

return (hayPaqueteConIdCódigo\_(idRepartidor))

}

function hayPaqueteConIdCódigo\_(idCodigo){

/\*

PROPÓSITO: Inidica si en la ubicación aactual hay un paquete con id de código \*\*idCodigo\*\*.

PARÁMETROS:

\*idCodigo: Número- el id del paquete que se busca el código

PRECONDICIONES:

\*Ninguna

TIPO: Booleano

\*/

return(hayPaqueteAcá() && hayCódigoAcá() && idCódigoAcá() == idCodigo)

}

//ejercicio 2

function mayorCantidadPaquetesSinEntregarDeUnRepartidor(){

/\*

PROPÓSITO: Describe la mayor cantidad de paquete sin entregar que correspondan a un repartidor.

PRECONDICIONES:

\*Debe haber una única cantidad máxima de paquetes por entregar de un repartidor.

TIPO: Número

\*/

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadPaquetes := máximoEntre\_Y\_(0, cantidadPaquetesPorEntregarDelRepartidorCeroSino())

while(haySiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadPaquetes := máximoEntre\_Y\_(cantidadPaquetes, cantidadPaquetesPorEntregarDelRepartidorCeroSino())

}

return(cantidadPaquetes)

}

function cantidadPaquetesPorEntregarDelRepartidorCeroSino(){

/\*

PROPÓSITO: Describe la cantidad de paquetes por entregar del repartidor de la ubicación actual cuando hay, cero cuando no hay.

PRECONDICIONES:

\*ninguna.

TIPO: Número

\*/

return(choose paquetesPorEntregar\_(idRepartidorAcá()) when (hayRepartidorAcá())

0 otherwise

)

}

function paquetesPorEntregar\_(idRepartidor){

/\*

PROPÓSITO: Describe la cantidad de paquetes por entregar del repartidor \*\*idRepartidor\*\*

PARÁMETROS:

\*idRepartidor: Número - el id del repartidor a contar los paquetes que faltan entregar

PRECONDICIONES:

\*ninguna.

TIPO: Número

\*/

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadPaquetes := unoSi\_ceroSino(hayPaqueteConIdCódigo\_(idRepartidor))

while(haySiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

cantidadPaquetes := cantidadPaquetes + unoSi\_ceroSino(hayPaqueteConIdCódigo\_(idRepartidor))

}

return (cantidadPaquetes)

}

//ejercicio 3

procedure EntregarPaquete\_(idPaquete){

/\*

PROPÓSITO: Entregar el paquete con id \*\*idPaquete\*\* al cliente que lo espera

PARÁMETROS:

\*idPaquete: Número - El id del paquete a ser entregado

PRECONDICIONES:

\*El cabezal se encuentra sobre el repartidor que debe entregar el paquete con id \*\*idPaquete\*\*

\*El paquete con id \*\*idPaquete\*\* se encuentra en el mapa

\*El cliente que espera el paquete se encuentra en el mapa

\*/

LlevarRepartidorAPaqueteConId\_(idPaquete)

LlevarPaqueteAlCliente()

}

procedure LlevarRepartidorAPaqueteConId\_(idPaquete){

/\*

PROPÓSITO: Llevar al repartidor de la ubicación actual al paquete con id \*\*idPaquete.

PRECONDICIONES:

\*El paquete con id \*\*idPaquete existe en el mapa y el mismo no fue entragado.

\*El cabezal se encuentra sobre el repartidor que debe entregar el paquete.

\*/

idRepartidor := idRepartidorAcá()

SacarRepartidor() //Sacar\_DeColor\_(idRepartidorAca(), repartidor())

IrAlPaquete\_(idPaquete)

PonerRepartidor\_(idRepartidor)//Poner\_DeColor\_(idRepartidor, repartidor())

}

procedure IrAlPaquete\_(idPaquete){

/\*

PROPÓSITO: lleva el cabezal al paquete de id \*\*idPaquete\*\*

PARÁMETROS:

\*idPaquete : Número - el id del paquete a llevar el cabezal

PRECONDICIONES:

\*Debe estar en el mapa la ubicación del paquete de id \*\*idPaquete\*\*

\*/

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

while(not esPaqueteDeId\_(idPaquete)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

}

}

function esPaqueteDeId\_(idPaquete){

/\*

PROPÓSITO: Indica si hay un paquete con id \*\*idPaquete\*\* en la celda actual.

PRECONDICIONES:

\*Ninguna.

TIPO: Booleano

\*/

return(hayPaqueteAcá() && idPaqueteAcá() == idPaquete)

}

procedure LlevarPaqueteAlCliente(){

/\*

PROPÓSITO: Lleva el paquete de la ubicación actual al cliente que lo espera.

PRECONDIDICIONES:

\*El cliente se encuentra en el mapa y todavía no ha recibido el paquete.

\*El repartidor que debe entregar el paquete se encuentra en la ubicación actual.

\*El paquete a entregar está en la ubicación actual.

\*/

idRepartidor := idRepartidorAcá()

idPaquete := idPaqueteAcá()

SacarRepartidor()

SacarPaquete()

SacarCódigo()

IrAPrimeraCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

while(not esClienteId\_(idPaquete)){

IrASiguienteCeldaEnUnRecorridoAl\_Y\_(Este, Norte)

}

EntregarPaquete\_DelRepartidor\_(idPaquete,idRepartidor)

}

procedure EntregarPaquete\_DelRepartidor\_(idPaquete, idRepartidor){

/\*

PROPÓSITO: Entregar el paquete \*idPaquete\* del repartidor \*idRepartidor\* en la ubicación actual

PARÁMETROS:

\*idPaquete: Número - el id del paquete a entregar

\*idRepartidor: Número - el id del repartidor que entrega el paquete

PRECONDICIONES:

\*En la ubicación no debe haber un paquete

\*En la ubicación no debe haber un repartidor

\*/

PonerPaquete\_(idPaquete)

PonerRepartidor\_(idRepartidor)

}