

```

program{
    return(cantidadDeClientesSinEntregas())
}

function cantidadDeClientesSinEntregas(){
    EntregarTodosLosPedidos()
    IrAPrimeraCeldaEnUnRecorridoAL_Y_(Este, Norte)
    cantidadDeClientesSinEntregas := unoSi_CeroSiNo(esClienteSinEntrega())
    while(haySiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)){
        IrASiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)
        cantidadDeClientesSinEntregas := cantidadDeClientesSinEntregas +
        unoSi_CeroSiNo(esClienteSinEntrega())
    }
    return(cantidadDeClientesSinEntregas)
}

function unoSi_CeroSiNo(condicion){
    return(
        choose 1 when (condicion)
        0 otherwise)
}

function esClienteSinEntrega(){
    return(hayBolitas(Azul) && not hayBolitas(Negro))
}

procedure EntregarTodosLosPedidos(){
    IrAPrimeraCeldaEnUnRecorridoAL_Y_(Este, Norte)
    while(haySiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)){
        IrASiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)
        EntregarPedidoSiHay()
    }
}

```

```

procedure EntregarPedidoSiHay(){
    if(hayBolitas(Rojo) && not hayBolitas(Azul)){
        EntregarPedido()
    }
}

procedure EntregarPedido(){
    numeroDeRepartidor := nroBolitas(Rojo)
    LlevarAlRepartidorHastaElPedido(numeroDeRepartidor)
    numeroDePedido := nroBolitas(Negro)
    LlevarElPedidoHastaElClienteConNumero_(numeroDePedido, numeroDeRepartidor)
}

procedure IrAPrimeraCeldaEnUnRecorridoAL_Y_(dirPrincipal, dirSecundaria){
    IrAlBorde(opuesto(dirPrincipal))
    IrAlBorde(opuesto(dirSecundaria))
}

function haySiguienteCeldaEnUnRecorridoAL_Y_(dirPrincipal, dirSecundaria){
    return(puedeMover(dirPrincipal) || puedeMover(dirSecundaria))
}

```

//Practica 8 - Ejercicio 6.c

```

procedure IrASiguienteCeldaEnUnRecorridoAL_Y_(dirPrincipal, dirSecundaria){
    if(puedeMover(dirPrincipal)){
        Mover(dirPrincipal)
    }else{
        IrAlBorde(opuesto(dirPrincipal))
        Mover(dirSecundaria)
    }
}

procedure EncontrarRepartidorConId_(numeroDeRepartidor){
    IrAPrimeraCeldaEnUnRecorridoAL_Y_(Este, Norte)
}

```

```

    while(nroBolitas(Rojo) /= numeroDeRepartidor &&
haySiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)){
        IrASiguienteCeldaEnUnRecorridoAL_Y_(Este,Norte)
    }
}

procedure LlevarAlRepartidorHastaElPedido(numeroDeRepartidor){
    Sacar_DeColor_(numeroDeRepartidor,Rojo)

    IrAPrimeraCeldaEnUnRecorridoAL_Y_(Este, Norte)

    while(nroBolitas(Verde) /= numeroDeRepartidor &&
haySiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)){
        IrASiguienteCeldaEnUnRecorridoAL_Y_(Este,Norte)
    }

    Poner_DeColor_(numeroDeRepartidor,Rojo)
}

procedure LlevarElPedidoHastaElClienteConNumero_(numeroDePedido,
numeroDeRepartidor){
    Sacar_DeColor_(numeroDeRepartidor,Rojo)

    Sacar_DeColor_(numeroDePedido,Negro)

    Sacar_DeColor_(nroBolitas(Verde),Verde)

    IrAPrimeraCeldaEnUnRecorridoAL_Y_(Este, Norte)

    while(nroBolitas(Azul) /= numeroDePedido &&
haySiguienteCeldaEnUnRecorridoAL_Y_(Este, Norte)){
        IrASiguienteCeldaEnUnRecorridoAL_Y_(Este,Norte)
    }

    Poner_DeColor_(numeroDeRepartidor,Rojo)

    Poner_DeColor_(numeroDePedido,Negro)
}

```