PRÁCTICA 5

# // EJ 1 (HECHO)

## A)

**procedure PonerElDobleDe\_(colorAPoner) {**

**/\***

**PROPÓSITO: Poner bolitas del color \*\*colorAPoner\*\* en**

**una cantidad que sea el doble de las que hay del**

**color \*\*colorAImitar\*\* en la celda actual.**

**PARÁMETROS:**

**\* colorAPoner : Color - color del que se**

**pondrán bolitas.**

**EXPRESIONES: nroBolitas() - Numero: Expresion que varia segun las bolitas del color dado**

**PRECONDICIONES: Ninguna**

**\*/**

**repeat(nroBolitas(colorAPoner)) {**

**Poner(colorAPoner)**

**}**

**}**

## B)

**procedure MoverEnLaDirección\_(direcciónAMover) {**

**/\***

**PROPÓSITO: Poner bolitas del color \*\*colorAPoner\*\* en**

**una cantidad que sea el doble de las que hay del**

**color \*\*colorAImitar\*\* en la celda actual.**

**PARÁMETROS:**

**\* direcciónAMover : Dirección - Dirección en la que**

**se movera el cabezal.**

**PRECONDICIONES: Ninguna**

**\*/**

**repeat(2) {**

**Mover(opuesto(opuesto(direcciónAMover)))**

**}**

**}**

## C)

**procedure PonerElDobleDe\_(colorAPoner) {**

**/\***

**PROPÓSITO: Poner bolitas del color \*\*colorAPoner\*\* en**

**una cantidad que sea el doble de las que hay del**

**color \*\*colorAImitar\*\* en la celda actual.**

**PARÁMETROS:**

**\* colorAPoner : Color - color del que se**

**pondrán bolitas.**

**EXPRESIONES: nroBolitas() - Numero: Expresion que varia segun las bolitas del color dado**

**PRECONDICIONES: Ninguna**

**\*/**

**repeat(nroBolitas(siguiente(Azul))) {**

**Poner(siguiente(colorAPoner))**

**}**

**}**

## D)

**procedure PonerElDobleDe\_QueDe\_(colorAPoner, colorAImitar) {**

**/\***

**PROPÓSITO: Poner bolitas del color \*\*colorAPoner\*\* en**

**una cantidad que sea el doble de las que hay del**

**color \*\*colorAImitar\*\* en la celda actual.**

**PARÁMETROS:**

**\* colorAPoner : Color - color del que se**

**pondrán bolitas.**

**\* colorAImitar : Color - color del que se**

**mirará cuántas bolitas hay en la**

**celda actual.**

**PRECONDICIONES: Ninguna**

**\*/**

**repeat(2 \* nroBolitas(colorAImitar)) {**

**Poner(colorAPoner)**

**}**

**}**

# // EJ 2 (HECHO)

**procedure Mover\_SegúnColor\_(dirección,color) {**

**/\***

**Prop.: Mover el cabezal hacia el \*dirección\*, igual a la cantidad de bolitas del \*color\***

**Prec.: Debe haber celdas igual o mayor a la cantidad**

**de bolitas del \*color\* hacia el \*dirección\*.**

**\*/**

**repeat(nroBolitas(color)) {**

**Mover(dirección)**

**}**

**}**

# // EJ 3 (HECHO)

## A)

**procedure Poner\_SemillasAl\_(cantidadDeSemillas, direcciónAPoner){**

**/\***

**PROPÓSITO: Poner bolitas rojas igual a la \*\*cantidadDeSemillas\*\* en**

**la direccion \*\*direcciónAPoner\*\*, dejando el cabezal en la celda inicial**

**PARÁMETROS:**

**\* cantidadDeSemillas : Numero - numero de bolitas rojas que se**

**pondran en la celda lindante**

**\* direcciónAPoner : Direccion - Direccion en la que se**

**movera el cabezal para poner semillas**

**PRECONDICIONES: Debe haber almenos una celda al \*\*direcciónAPoner\*\***

**\*/**

**Mover(direcciónAPoner)**

**repeat(cantidadDeSemillas) {**

**Poner(Rojo)**

**}**

**Mover(opuesto(direcciónAPoner))**

**}**

## B)

**procedure Sacar\_ÁrbolesAl\_(cantidadDeÁrboles, direcciónASacar) {**

**/\***

**PROPÓSITO: Sacar bolitas verdes igual a la \*\*cantidadDeÁrboles\*\* en**

**la direccion \*\*direcciónASacar\*\*, dejando el cabezal en la celda inicial**

**PARÁMETROS:**

**\* cantidadDeÁrboles : Numero - numero de bolitas verdes que se**

**sacaran en la celda lindante**

**\* direcciónASacar : Direccion - Direccion en la que se**

**movera el cabezal para sacar Arboles**

**PRECONDICIONES: Debe haber almenos una celda al \*\*direcciónASacar\*\* y bolitas verdes**

**igual o mayor \*\*cantidadDeÁrboles\*\***

**\*/**

**Mover(direcciónASacar)**

**repeat(cantidadDeÁrboles) {**

**Sacar(Verde)**

**}**

**Mover(opuesto(direcciónASacar))**

**}**

## C)

**procedure Sacar\_SemillasEnDiagonalAl\_Y\_(cantidadDeSemillas, primeraDirDiagonal, segundaDirDiagonal){**

**/\***

**PROPÓSITO: Sacar bolitas Roja igual a la \*\*cantidadDeSemillas\*\* en**

**la direccion \*\*primeraDirDiagonal\*\* y \*\*segundaDirDiagonal\*\*, dejando**

**el cabezal en la celda inicial**

**PARÁMETROS:**

**\* cantidadDeSemillas : Numero - numero de bolitas verdes que se**

**sacaran en la celda lindante**

**\* primeraDirDiagonal : Direccion - La primer dirección en la que se**

**movera el cabezal para sacar Semillas**

**\* segundaDirDiagonal : Direccion - La segunda dirección en la que se**

**movera el cabezal para sacar Semillas**

**PRECONDICIONES: Debe haber almenos una celda al \*\*primeraDirDiagonal\*\* y \*\*segundaDirDiagonal\*\* y bolitas Rojas**

**igual o mayor \*\*cantidadDeSemillas\*\***

**\*/**

**Mover(primeraDirDiagonal)**

**Mover(segundaDirDiagonal)**

**repeat(cantidadDeSemillas) {**

**Sacar(Rojo)**

**}**

**Mover(opuesto(primeraDirDiagonal))**

**Mover(opuesto(segundaDirDiagonal))**

**}**

## 

## D)

**procedure Sacar\_ÁrbolesEnDiagonalHorariaAl\_(cantidadDeÁrboles, direcciónDiagonal){**

**/\***

**PROPÓSITO: Sacar bolitas Roja igual a la \*\*cantidadDeSemillas\*\* en**

**la direccion \*\*primeraDirDiagonal\*\* y \*\*segundaDirDiagonal\*\*, dejando**

**el cabezal en la celda inicial**

**PARÁMETROS:**

**\* cantidadDeÁrboles : Numero - numero de bolitas verdes que se**

**sacaran en la celda lindante**

**\* direcciónDiagonal : Direccion - La dirección diagonal en la que se**

**movera el cabezal para sacar Arboles**

**PRECONDICIONES: Debe haber almenos una celda al \*\*direcciónDiagonal\*\* y bolitas Rojas**

**igual o mayor \*\*cantidadDeÁrboles\*\***

**\*/**

**Mover(direcciónDiagonal)**

**repeat(cantidadDeÁrboles) {**

**Sacar(Verde)**

**}**

**Mover(opuesto(direcciónDiagonal))**

**}**

# // EJ 4

# // EJ 5 (HECHO)

## A)

**procedure EnviarAliadosParaDuplicarEnemigos() {**

**/\***

**Prop.: Agrega soldados aliados en cantidad equivalente al doble**

**de bolitas de soldados enemigos**

**Prec.: Ninguna**

**Obs.: Los aliados se representan con bolitas negras y los enemigos**

**con bolitas rojas**

**\*/**

**Poner\_DeColor\_((nroBolitas(Rojo)\*2) - nroBolitas(Negro), Negro)**

**}**

## B)

**procedure PelearLaBatalla() {**

**/\***

**Prop.: Simular una batalla, suponiendo que hay soldados aliados como para ganar la batalla**

**contra 3 soldados aliados y todos mueren**

**Prec.: Hay mas aliados que enemigos, en proporcion. Es decir, 3 aliados**

**por cada 2 enemigos**

**Obs.: En el diseño de esta batalla siempre ganan los aliados**

**\*/**

**Sacar\_DeColor\_(3 \* (nroBolitas(Rojo)) div 2, Negro)**

**SacarTodasLasDeColor\_(Rojo)**

**}**

# // EJ 6 (HECHO)

## BIBLIOTECA

**procedure SacarTodasLasDeColor\_(colorASacar) {**

**/\***

**PROPÓSITO: Saca todas las bolitas del color \*\*colorASacar\*\* de la celda actual.**

**PARÁMETROS:**

**\* colorASacar : Color - El color de las bolitas a sacar.**

**PRECONDICIÓN: Ninguna.**

**\*/**

**repeat(nroBolitas(colorASacar)) {**

**Sacar(colorASacar)**

**}**

**}**

# // EJ 7 (HECHO)

## BIBLIOTECA

**procedure VaciarCelda() {**

**/\***

**Prop.: Vacía la celda de todas las bolitas de colores**

**Prec.: Ninguna**

**\*/**

**repeat(nroBolitas(Azul)) {**

**Sacar(Azul)**

**}**

**repeat(nroBolitas(Negro)) {**

**Sacar(Negro)**

**}**

**repeat(nroBolitas(Rojo)) {**

**Sacar(Rojo)**

**}**

**repeat(nroBolitas(Verde)) {**

**Sacar(Verde)**

**}**

**}**

# // EJ 8

# // EJ 9 (HECHO)

**procedure PasarPalabraActualAMayúsculas(){**

**/\***

**Prop.: Pasa una palabra escrita en minúscula en la fila actual a mayúscula y dejandola en la fila lindante**

**al Norte de la actual.**

**Prec.: \* El cabezal se encuentra en la celda más al Oeste de la palabra.**

**\* Debe haber al menos tantas celdas al Este de la celda actual como tenga codificada.**

**\* Debe haber al menos una fila lindante al Norte de la actual y las celdas sobre la palabra estan vacias**

**Obs.: \* Una palabra comienza por una celda con bolitas de color Rojo cuya cantidad**

**indica la cantidad de letras que posee la palabra.**

**\* La primer celda al Este desde el comienzo, y sus subsiguientes, contienen las**

**letras de la palabra, dadas por la cantidad de bolitas de color Negro.**

**\* Cada letra se codifica siguiendo el código ASCII.**

**\*/**

**repeat(nroBolitas(Rojo)){**

**PasarALaSiguienteLetra() //Mover(Este)**

**EscribirLetraEnMayúsculasAlNorte()**

**}**

**IrAlBorde(Oeste)**

**}**

**procedure PasarALaSiguienteLetra(){**

**/\***

**Prop.:Mueve el cabezal a la próxima letra de una palabra.**

**Prec.: \* El cabezal se encuentra en la celda de una palabra (Ya sea al comienzo o**

**sobre alguna letra).**

**\* Hay al menos una celda al Este de la actual.**

**\*/**

**Mover(Este)**

**}**

**procedure EscribirLetraEnMayúsculasAlNorte(){**

**/\***

**Prop.: Escribe la letra sobre la cual se encuentra el cabezal en la celda**

**lindante al Norte en mayúsculas.**

**Prec.: El cabezal se encuentra sobre una letra y esta está en minuscula.**

**Debe existir al menos una celda lindante al norte de la actual y debe estar vacia**

**obs.: \* Cada letra se codifica siguiendo el código ASCII con bolitas negras.**

**\*/**

**EscribirLetra\_Al\_(nroBolitas(Negro)-32, Norte)**

**}**

**procedure EscribirLetra\_Al\_(codigoASCII, direccion){**

**/\***

**Prop.: Escribe la letra con código ASCII \*\*códigoASCII\*\* en la celda lindante**

**en dirección \*\*dirección\*\*.**

**Parametros: \*codigoASCII\* - Numero - codigo ASCII de la letra**

**\*direccion\* - Direccion - direccion donde se escribe la letra**

**Prec.: Debe existir al menos una celda lindante al \*direccion\* de la actual y debe estar vacia**

**\*/**

**Mover(direccion)**

**Poner\_DeColor\_(codigoASCII, Negro)**

**Mover(opuesto(direccion))**

**}**

**/\* Debe haber al menos \*\*ancho - 1\*\* celdas hacia el Este de la celda actual.**

**\* Debe haber al menos \*\*ancho + alto - 2\*\* celdas hacia el Norte de la celda actual.\*/**