

Práctica 4

Repetición condicional y recorridos

Introducción a la Programación 2018 S2

Repetición condicional

Ejercicio 1

Corresponder las siguientes oraciones que describen propósitos y precondiciones con sus respectivos procedimientos.

1. Mueve el cabezal al extremo en dirección d.
2. No tiene precondición.
3. Posiciona el cabezal en alguna celda de la fila actual que no tenga bolitas.
4. Hay alguna celda con bolitas en la fila actual
5. Dibuja una diagonal de color c en el tablero.
6. Hay mas filas que columnas en el tablero.
7. Hay mas columnas que filas en el tablero.

```
procedure A(d) {
    while(puedeMover(d)) {
        Mover(d)
    }
}
procedure B() {
    A(Oeste)
    while(conBolitas()) { //con conBolitas() es una funcion que devuelve Verdadero
        //si en la celda actual hay al menos una bolita
        Mover(Este)
    }
}
procedure D(c) {
    IrAlOrigen()
    while(puedeMover(Este)) {
        Poner(c) Mover(Este) Mover(Norte)
    }
    Poner(c)
}
procedure E(c) {
    IrAlOrigen()
    while(puedeMover(Norte)) {
        Poner(c)
        Mover(Este)
        Mover(Norte)
    }
    Poner(c)
}
```

```

procedure F(c) {
    IrAlOrigen()
    while(puedeMover(Este) && puedeMover(Norte)) {
        Poner(c)
        Mover(Este)
        Mover(Norte)
    }
    Poner(c)
}

```

Ejercicio 2

Determine las precondiciones de los siguientes procedimientos.

```

procedure MayorarHasta(c1, c2) {
    //Pone bolitas de color c1 para igualar la cantidad de bolitas de color c2
    while(nroBolitas(c1) != nroBolitas(c2)) {
        Poner(c1)
    }
}

```

```

procedure RobinHoodear(c1, c2) {
    //Saca del que mas color hay para poner del que menos color hay hasta igualar
    while(nroBolitas(c1) != nroBolitas(c2)) {
        if(nroBolitas(c1) >nroBolitas(c2)) {
            Sacar(c1); Poner(c2);
        }
        else { Sacar(c2); Poner(c1);
        }
    }
}

```

```

procedure PingPonguear(c1, c2) {
    //Mueve el cabezal en forma de ping pong de acuerdo a las bolitas de color c1 y c2
    while(hayBolitas(c1) || hayBolitas(c2)) {
        if(hayBolitas(c1)) {
            MoverN(nroBolitas(c1), Este)
        }
        else {
            MoverN(nroBolitas(c2), Oeste);
        }
    }
}

```

Ejercicio 3

Escriba el procedimiento `IrHastaColor(color, dir)` que posicione el cabezal en la primer celda en dirección `dir` que contenga bolitas de color "color".

Ejercicio 4

Escriba un procedimiento `PintarConColorHacia(color, dir)` que coloque una bolita de color "color" en todas las celdas que aparecen en dirección `dir` desde la celda actual.

Ejercicio 5

Escriba un procedimiento `IrAlExtremo(dir)` que posicione el cabezal en la última celda en un recorrido en dirección `dir`.

Ejercicio 6

Usando `IrAlExtremo`, escriba el procedimiento `IrAlOrigenBis()` que posicione el cabezal en el origen del tablero.

Ejercicio 7

Combine `IrAlExtremo` y `PintarConColorHacia` para obtener el procedimiento `PintarFila(color)` que coloque una bolita de color "color" en cada una de las celdas de la fila actual.

Ejercicio 8

Escribir un procedimiento `IrAlUltimaCeldaDeLaFilaAlSinBolitas(dir)` que deje el cabezal en la celda que se encuentre más cercana al extremo `dir` de la fila actual y que no tenga ninguna bolita. Cual es la precondición?

Recorridos:

Ejercicio 9:

Escribir las siguientes funciones y procedimientos que sirven para recorrer el tablero.

El recorrido se hace avanzando en la dirección `dirPrincipal` y cuando no se pueda volviendo al extremo y siguiendo en la `dirSecundaria`.

Por ejemplo, en la siguiente Figura se muestra el recorrido cuando `dirPrincipal` denota Sur y `dirSecundaria` denota Oeste; el recorrido empieza en el extremo Noreste, y se va avanzando por columnas hacia el Oeste, recorriendo cada columna de Norte a Sur.

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

1. InicializarRecorridoHacia(dirPrincipal ,dirSecundaria), que recibe dos direcciones dirPrincipal y dirSecundaria y mueve el cabezal al inicio del recorrido. En otras palabras, el cabezal se mueve al extremo opuesto(d1) y opuesto(dirSecundaria).

2. hayProximaCelda(dirPrincipal , dirSecundaria) que denota Verdadero cuando el cabezal se puede avanzar a la siguiente celda. En otras palabras, denota Verdadero cuando el cabezal no se encuentra en el extremo dirPrincipal y extremo dirSecundaria.

3. IrAProximaCelda(dirPrincipal , dirSecundaria) que mueve el cabezal a la siguiente celda del recorrido. En otras palabras, el cabezal se ubica en la celda lindante en dirección dirPrincipal de ser posible. Si no es posible, entonces el cabezal se mueve en dirección dirSecundaria y hacia el extremo opuesto(dirPrincipal). ¿Cuál es la precondition de este procedimiento?

Ejercicio 10:

Identifique el esquema de recorrido de las siguientes procedimientos, indicando qué elementos se recorren (e.g., las celdas del tablero, las celdas de una fila, las celdas de una columna, las celdas que tienen tales bolitas, etc.), los procedimientos que se usan para ir a la primer celda, avanzar a la siguiente celda y procesar cada celda, y la función que se utiliza para determinar si hay más celdas por procesar dentro del recorrido.

Escriba, asimismo, las condiciones de los procedimientos.

```
procedure LimpiarTablero() {
    InicializarRecorridoHacia(Norte, Este)
    while( hayProximaCelda(Norte, Este)) {
        LimpiarCelda()
        MoverT(Norte, Este)
    }
    LimpiarCelda()
}
```

```
procedure LimpiarFila() {
    IrAlExtremo(Oeste)
    while(puedeMover(Este)) {
        LimpiarCelda()
        Mover(Este)
    }
    LimpiarCelda() }
```

```
procedure IrAVaciaHacia(d) {
    while(puedeMover(d) && not vacia()) {
        Mover(d)
    }
}
```

```

procedure IrASiguienteColor(c) {
    while(not hayBolitas(c)) {
        IrAProximaCelda(Norte, Este)
    }
}

```

```

procedure IrASiguienteColor(c) {
    while( hayProximaCelda(Norte, Este) && not hayBolitas(c)) {
        IrAProximaCelda(Norte, Este)
    }
}

```

```

procedure IrAUltimaConColor(c) {
    while(haySiguieteColor(c)) {
        IrASiguienteColor(c)
    }
}

```

Ejercicio 11

Escribir un procedimiento PonerEnVacias(color) que ponga una bolita de color "color" en todas las celdas vacías del tablero. El procedimiento debe utilizar un esquema de recorrido que recorra todas las celdas del tablero.

Ejercicio 12 Escribir un procedimiento DegradarTablero(color,deg) que, dado un color "color" y un número deg, haga lo siguiente para cada celda del tablero. Si hay más de deg bolitas de color "color" en una celda, entonces el procedimiento saca deg bolitas de la celda; caso contrario, saca todas las bolitas de color "color".

Ejercicio 13

Escribir el procedimiento BuscarSiguieteVacía(dirPrincipal,dirSecundaria) que posicione el cabezal en la siguiente celda vacía del tablero en un recorrido en direcciones dirPrincipa y dirSecundaria

(ver Ejercicio 9). Puede suponer que dicha celda siempre existe.

Ejercicio 14

Escribir un procedimiento RecorrerSegunColor(color) que realice la siguiente acción: comienza moviéndose hacia el Norte tantas celdas como bolitas de color "color" haya en la celda actual, una vez ahí repite la acción (tomando la cantidad de bolitas de color "clor" de la nueva celda actual). Continuar moviendose cambiando la la cantidad de celdas hasta caer en una celda que no tenga bolitas de color "clor". Sugerencia: utilizar un recorrido. ¿Que precondition tiene el procedimiento RecorrerSegunColor(color)?.