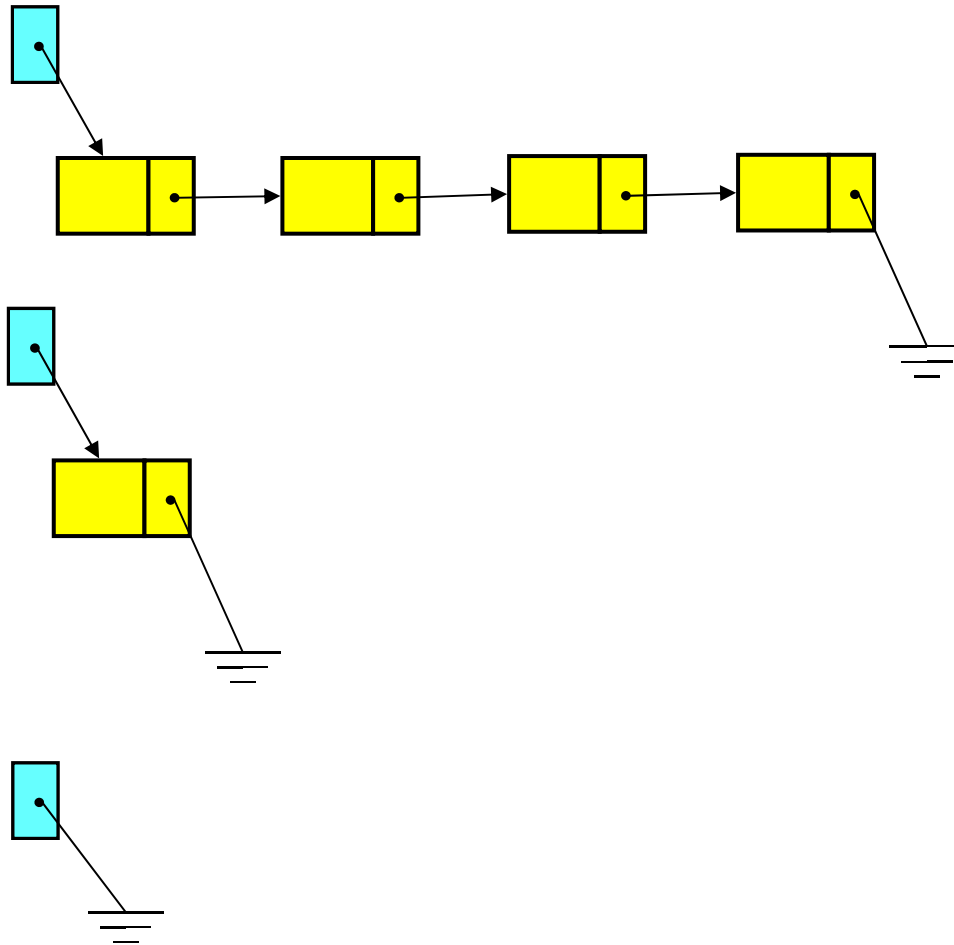
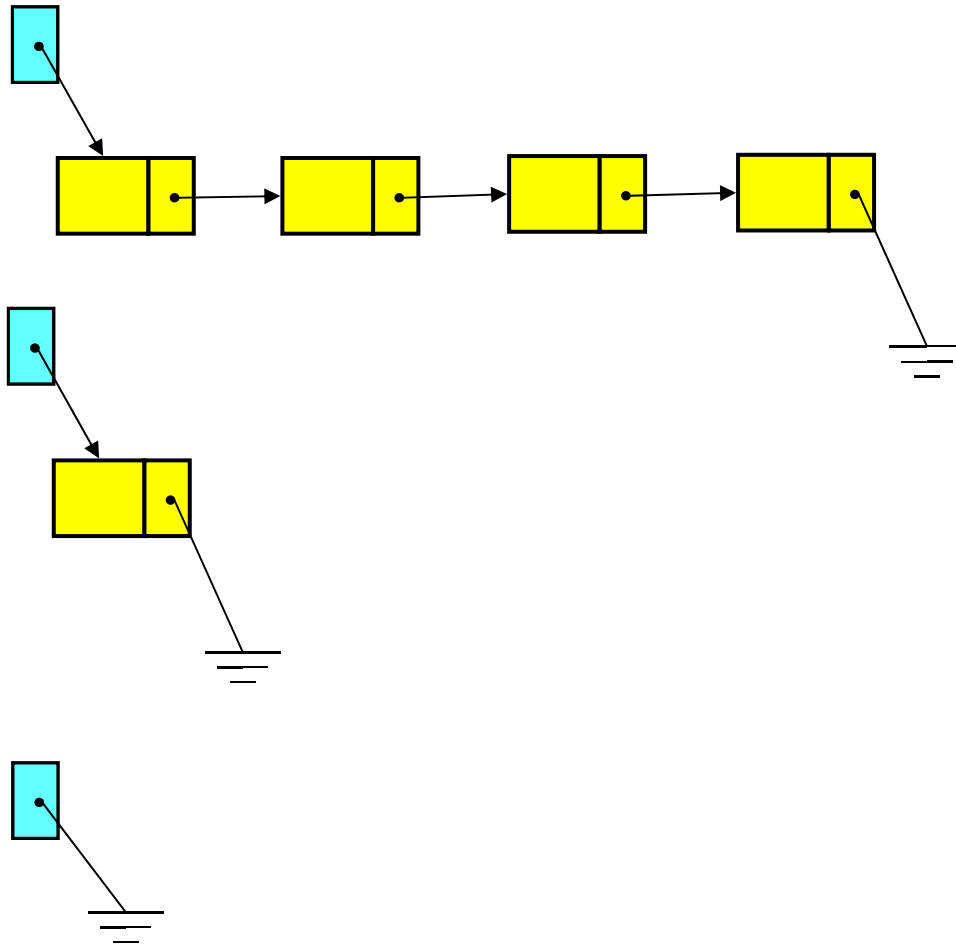


Lista Simplemente Ligada Lineal Sin Encabezado (LSLLSE)

Sin encabezado





```

typedef _____ tipo_dato;

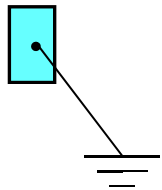
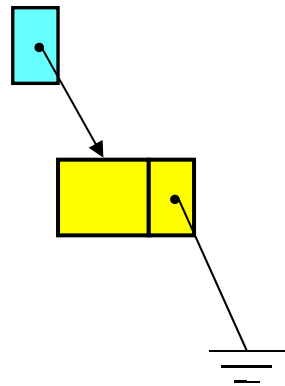
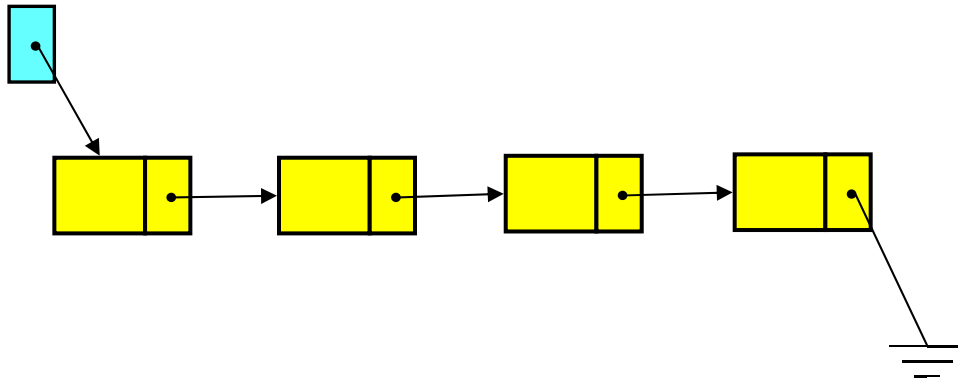
typedef struct tipo_nodo {
    tipo_dato elem;
    struct tipo_nodo *sig;
} tipo_nodo;

typedef tipo_nodo *tipo_lista;

typedef tipo_nodo *tipo_pos;

int main() {
    tipo_lista r;
    r = X NULL;
    inicializa (&r);
    .
    .
}

```

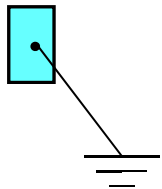
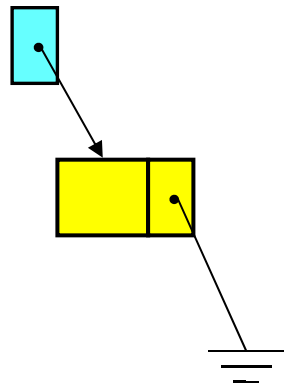
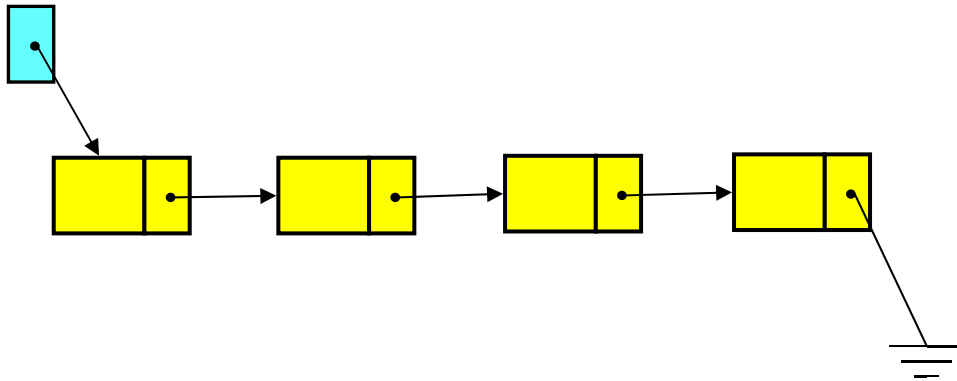


función: **inicializa**

recibe: *lista*

regresa: nada

lista = NULO



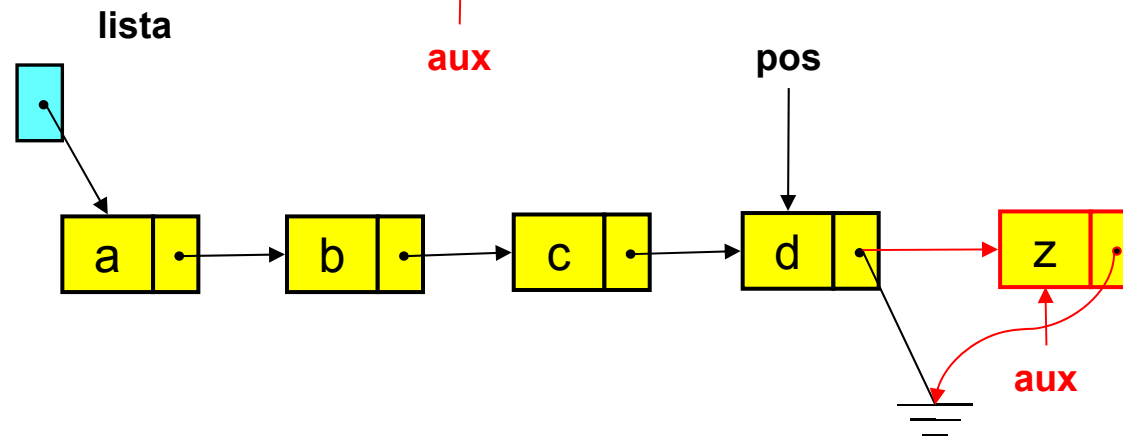
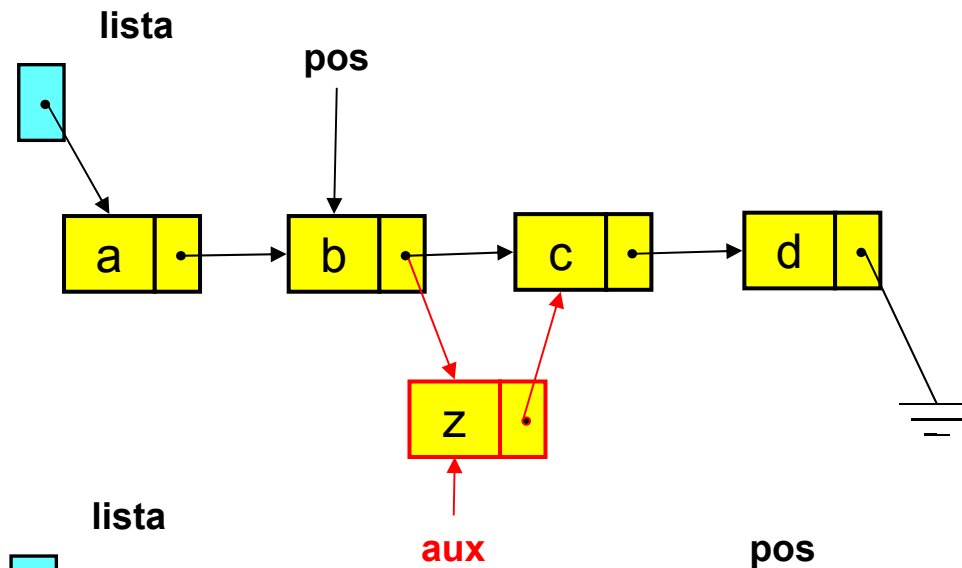
función: **vacía**
 recibe: *lista*
 regresa: booleano

¿lista = NULO?

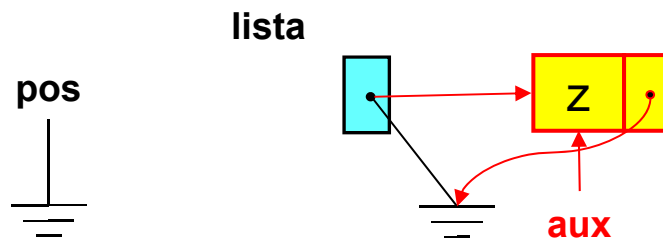
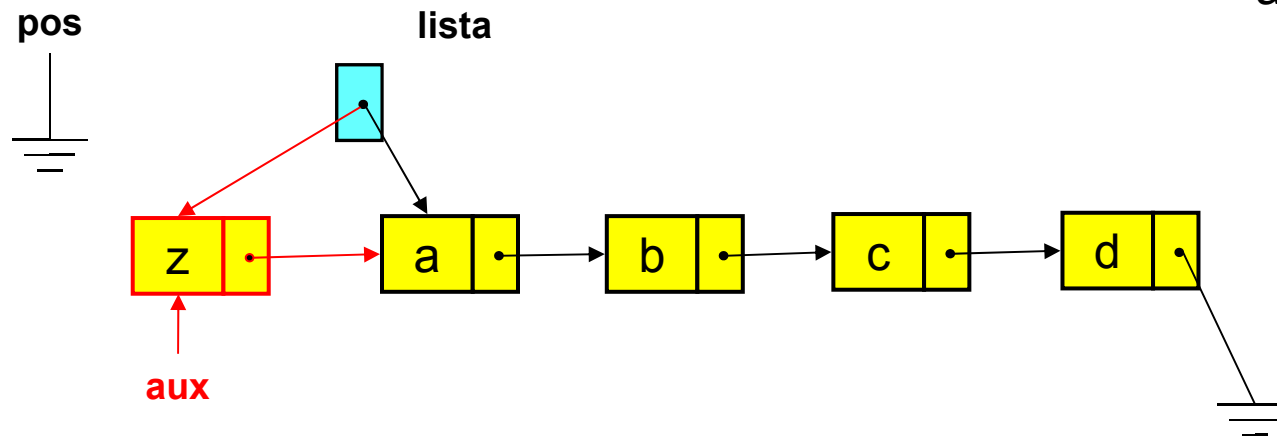
Sí: regresar: *verdadero*

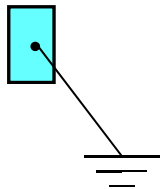
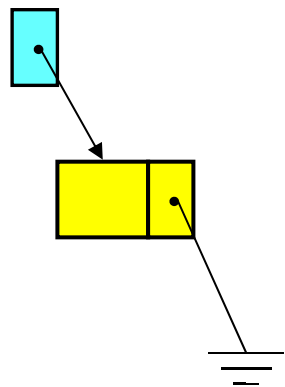
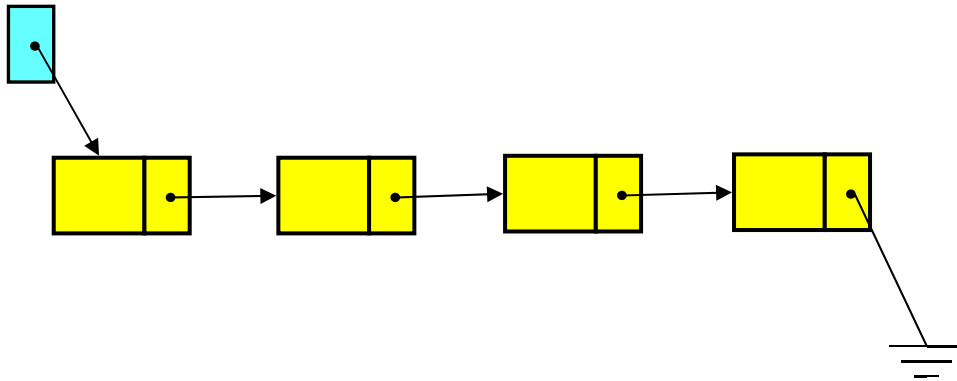
No: regresar: *falso*

Inserción
en cualquier posición,
excepto al principio



Inserción
al principio





función: **inserta**

recibe: *elem*, *pos*, *lista*

regresa: nada

aux = nuevo nodo

aux→*elem* = *elem*

¿*pos* = NULO?

Sí: *aux*→*sig* = *lista*
 lista = *aux*

No: *aux*→*sig* = *pos*→*sig*
 pos→*sig* = *aux*

```

void inicializa (tipo_Lista *L) {
    *L = NULL;
}

```

```

int vacia (tipo_Lista *L) {
    if (*L == NULL)
        return TRUE;
    else
        return FALSE;
}

```

```

int vacia (tipo_Lista *L) {
    return *L == NULL;
}

```

```

int vacia (tipo_Lista *L) {
    return !*L;
}

```

```

void inserta (tipo_Dato e, tipo_Pos pos, tipo_Lista *L) {
    tipo_Pos aux;

    aux = (tipo_Pos)malloc (sizeof (tipo_Nodo));

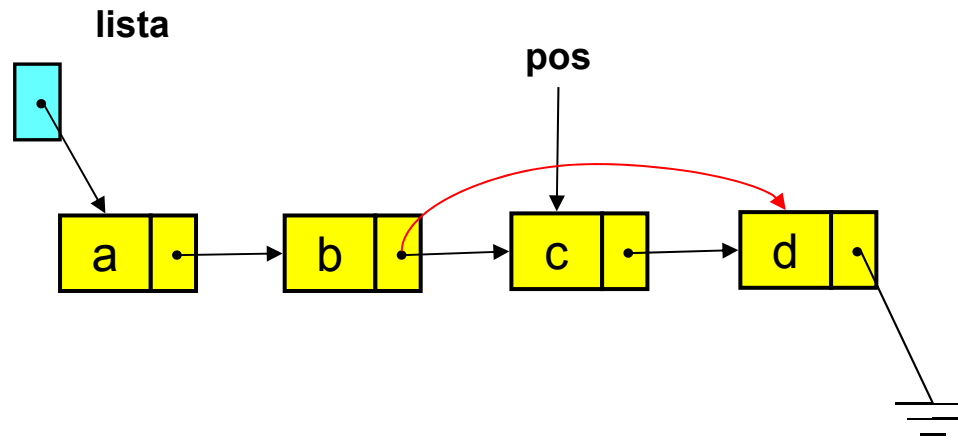
    if (aux == NULL) {
        printf ("memoria insuficiente");
        return;
    }

    aux->elem = e;

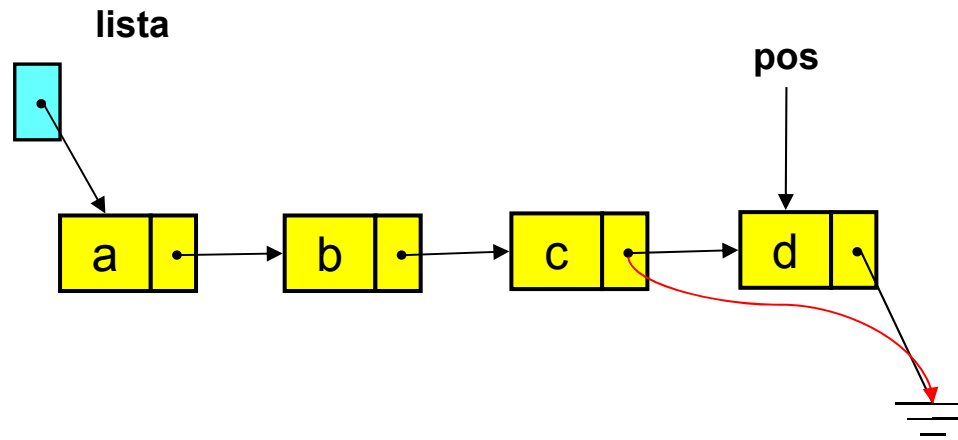
    if (pos == NULL) {
        aux->sig = *L;
        *L = aux;
    }
    else {
        aux->sig = pos->sig;
        pos->sig = aux;
    }
}

```

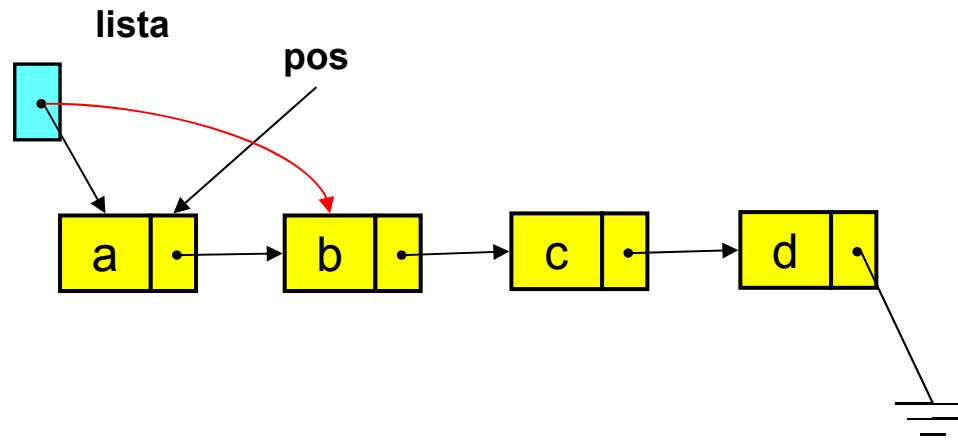
Eliminación
en cualquier posición,
excepto al principio



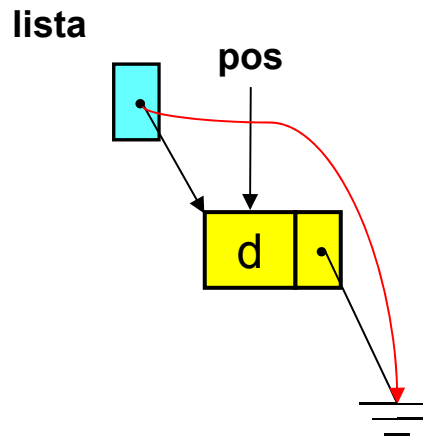
Eliminación
en cualquier posición,
excepto al principio

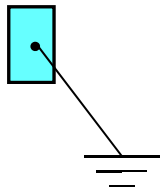
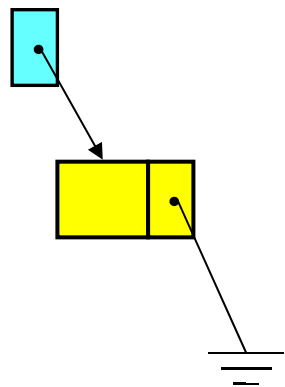
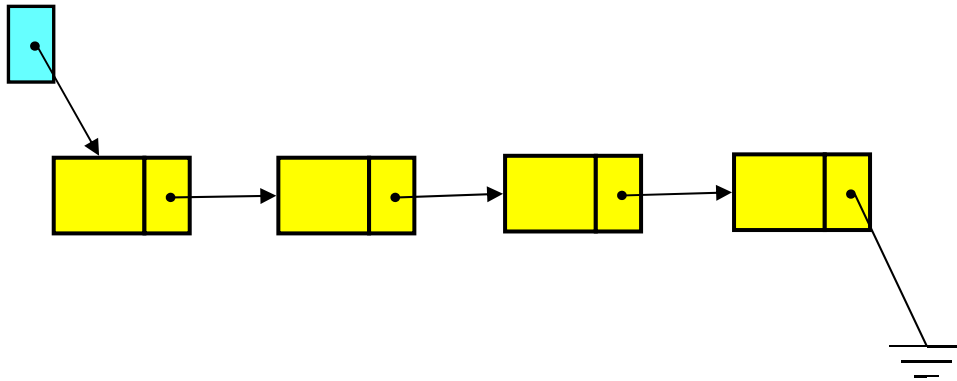


Eliminación
al principio



Eliminación al principio





función: **elimina**

recibe: *pos*, *lista*

regresa: nada

¿**vacía**(*lista*) ó *pos* = NULO?

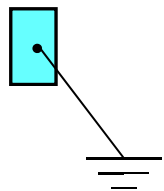
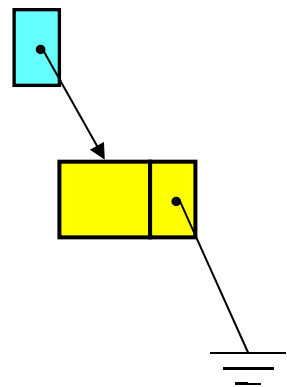
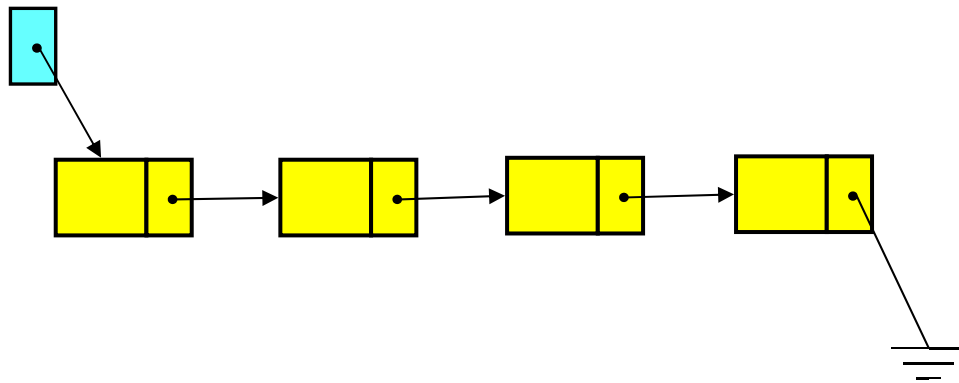
Sí: terminar

¿*pos* = *lista*?

Sí: $lista = lista \rightarrow sig$

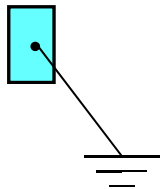
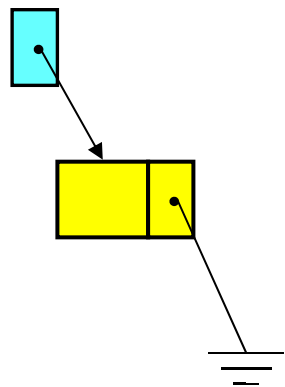
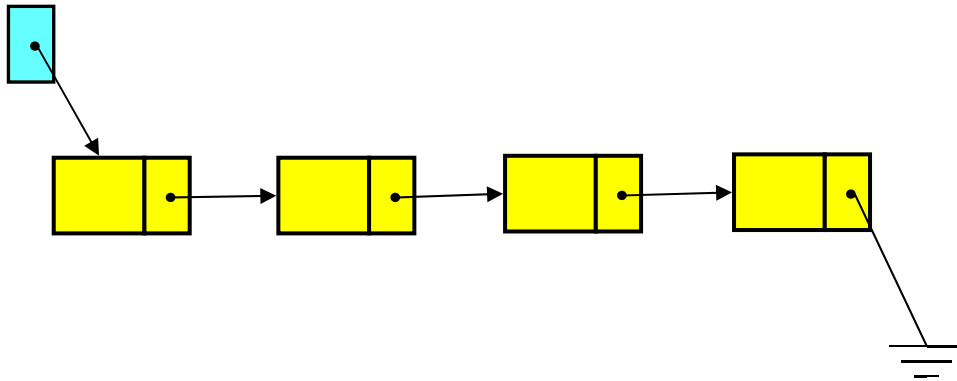
No: **anterior**(*pos*, *lista*) $\rightarrow sig = pos \rightarrow sig$

liberar espacio de memoria de *pos*



función: ***primero***
recibe: *lista*
regresa: posición

regresar: *lista*



función: **último**

recibe: *lista*

regresa: posición

¿vacía(*lista*)?

Sí: regresar: NULO

terminar

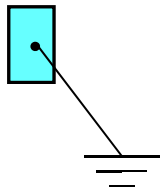
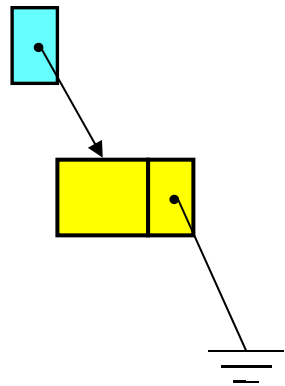
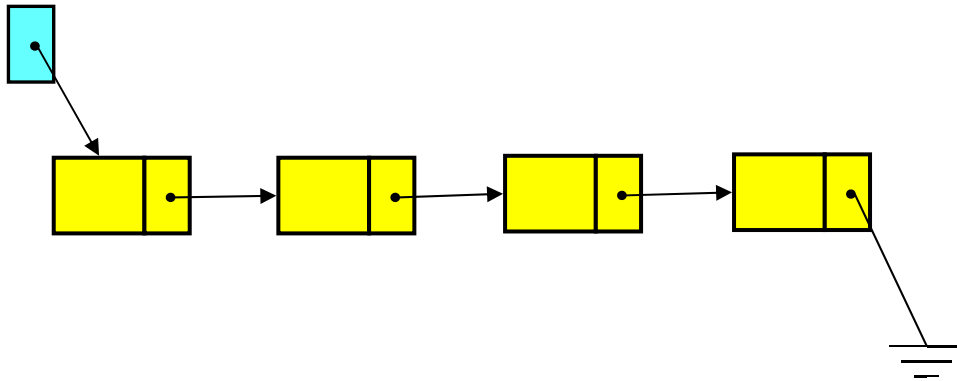
aux = *lista*

mientras *aux*→sig ≠ NULO

aux = *aux*→sig

fin mientras

regresar: *aux*



función. **Anterior**

recibe: *pos*, *lista*

regresa: posición

¿vacía(*lista*) ó *pos* = NULO?

Sí: regresar: NULO

terminar

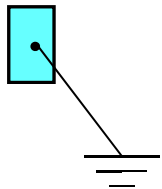
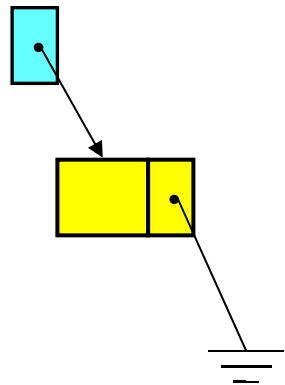
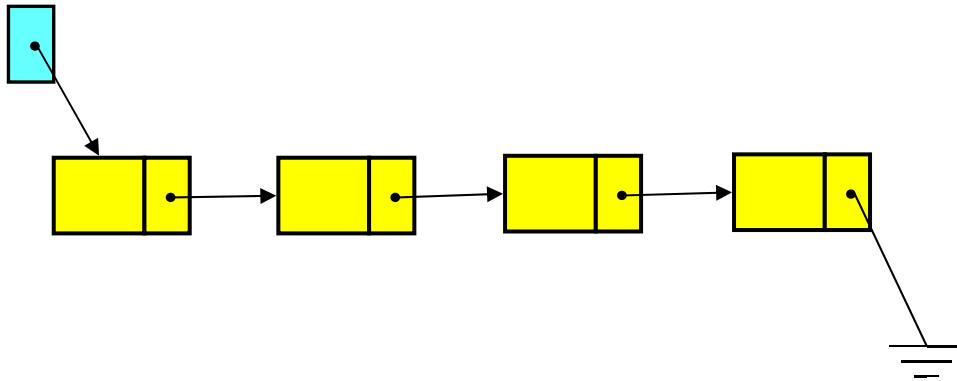
aux = *lista*

mientras *aux* ≠ NULO y *aux*→sig ≠ *pos*

aux = *aux*→sig

fin mientras

regresar: *aux*



función: **siguiente**

recibe: *pos*, *lista*

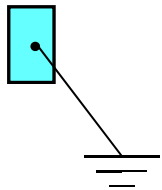
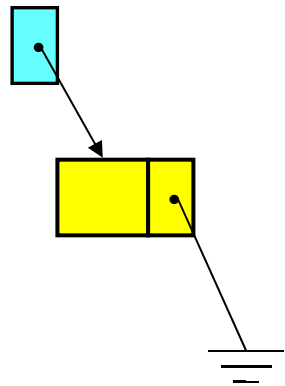
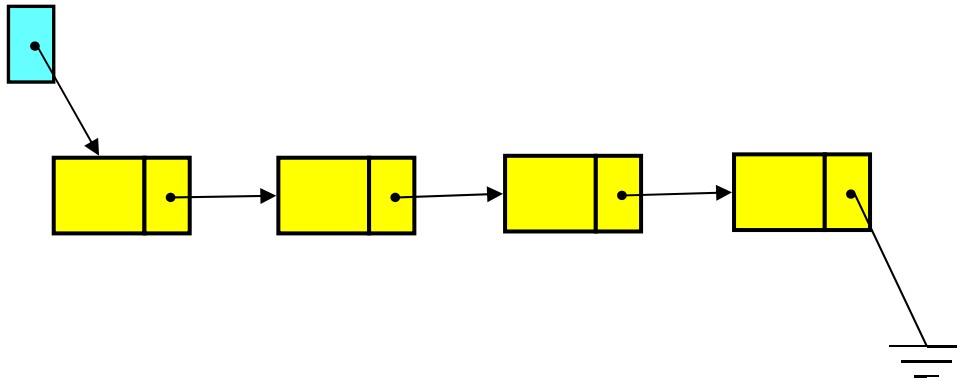
regresa: posición

¿vacía(*lista*) ó *pos* = NULO?

Sí: regresar: NULO

terminar

regresar: *pos* → sig



función: **localiza**

recibe: *elem*, *lista*

regresa: posición

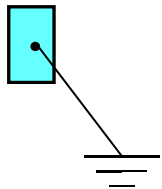
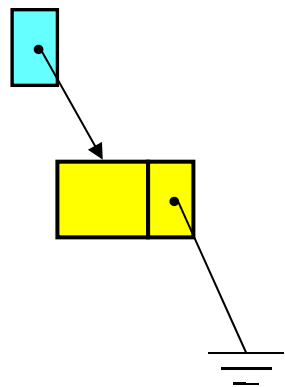
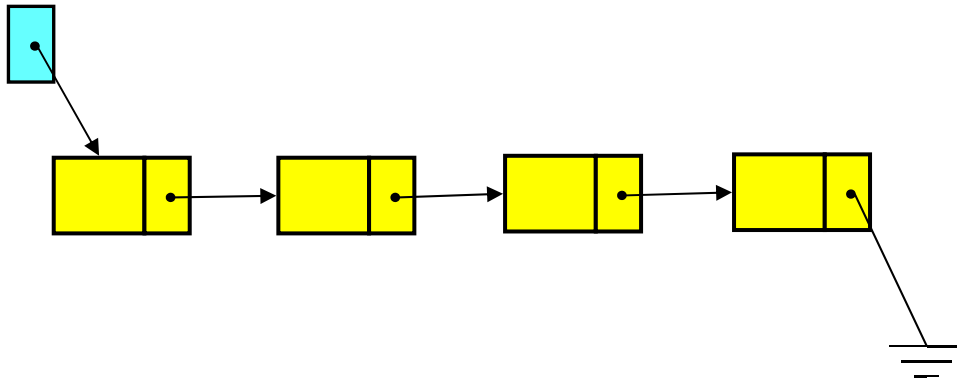
aux = *lista*

mientras *aux* ≠ NULO y *aux*→*elem* ≠ *elem*

aux = *aux*→*sig*

fin mientras

regresar: *aux*



función: **recupera**

recibe *pos*, *lista*

regresa: elemento

¿vacía(*lista*) ó *pos* = NULO?

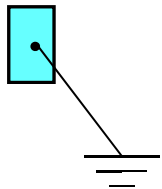
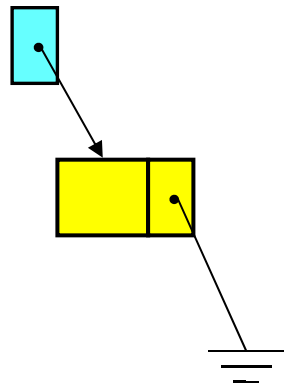
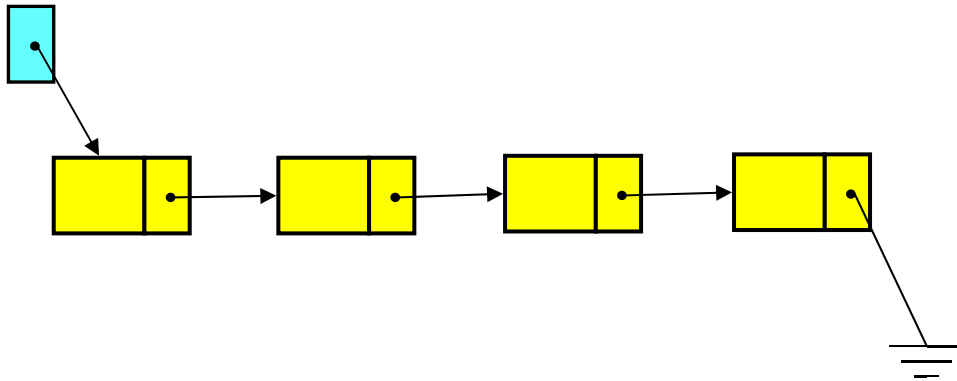
Sí: ¡error de excepción!

Insuficiencia de datos

terminar



No: regresar: *pos*→elem



función: **imprime**

recibe: *lista*

regresa: nada

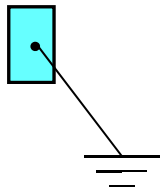
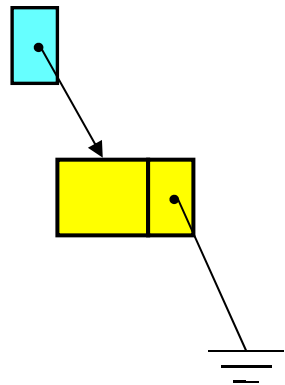
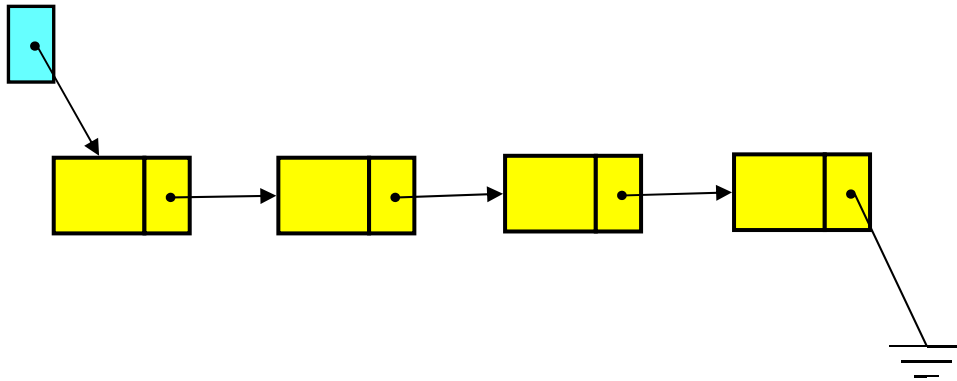
aux = *lista*

mientras *aux* ≠ NULO

 imprimir *aux*→elem

aux = *aux*→sig

fin mientras



función: **anula**

recibe: *lista*

regresa: nada

mientras $lista \neq \text{NULO}$

$aux = lista$

$lista = lista \rightarrow sig$

liberar espacio de memoria de aux

fin mientras

