

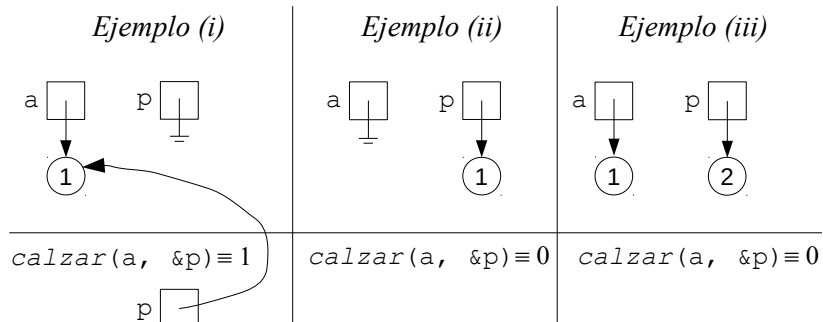
Considere la siguiente definición para el tipo *Nodo*:

```
typedef struct nodo {
    int x;
    struct nodo *izq, *der;
} Nodo;
```

Programa la función:

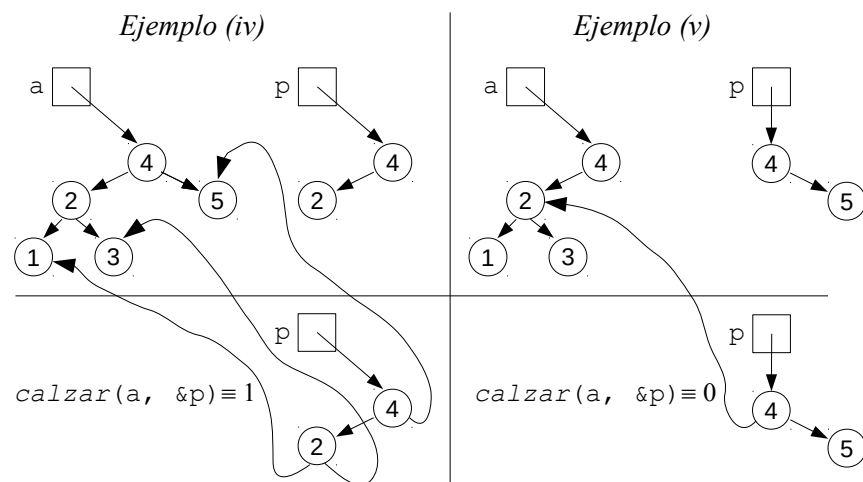
```
int calzar(Nodo *a, Nodo **ppat);
```

Esta función trata de calzar el patrón almacenado en *\*ppat* con el árbol binario *a*. El patrón también es un árbol binario. Calzar el patrón significa modificar los punteros nulos en el patrón de manera que apunten a nodos del árbol *a* hasta que el patrón sea igual a *a* (vea el ejemplo iv). En tal caso la función entrega 1 (verdadero). Si no se puede calzar el patrón la función retorna 0 (falso). En los siguientes ejemplos la celda superior señala la situación inicial mientras que la celda inferior indica la llamada realizada, el valor retornado y cómo se modifica el patrón.



El ejemplo (i) muestra que un patrón vacío calza con cualquier árbol, en cuyo caso el patrón se modifica de modo que sea el árbol *a* (que puede ser vacío también). El ejemplo (ii) indica que un árbol vacío no calza con un patrón no vacío y el patrón no se modifica. El ejemplo (iii) señala que un patrón no calza con un árbol cuando sus raíces difieren (el valor del campo *x* difiere). El patrón no se modifica.

El ejemplo (iv) arriba a la derecha muestra que un patrón no vacío calza con un árbol no vacío con igual raíz cuando ambos subpatrones calzan con sus respectivos subárboles. En tal caso se modifican los subpatrones. Por último el ejemplo (v) señala que el patrón no calza cuando uno de sus subpatrones no calza, pero de todos modos se modifica el otro subpatrón si éste calza con su respectivo subárbol.



*Ayuda:* Los primeros 3 ejemplos corresponden a casos base (no hay recursividad). Los últimos 2 ejemplos son los casos recursivos.

### Recursos

Baje *t2.zip* de U-cursos y descomprímalo. El directorio *T2* contiene los archivos *test-t2.c* que prueba si su tarea funciona, *t2.h* que incluye los encabezados de las funciones pedidas y *Makefile*. Ud. debe programar la función pedida en el archivo *t2.c*. Lea las instrucciones que aparecen al comienzo de *Makefile*. Ud. debe usar el comando *make* para compilar su tarea. El programa de prueba lo felicitará si su tarea aprueba todos los tests o le indicará cuál test falla.

### Entrega

Ud. solo debe entregar el archivo *t2.c* por medio de U-cursos. Entregue su tarea solo si compila sin arrojar warnings en la máquina *anakena.dcc.uchile.cl* y aprueba todos los tests. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo o festivos.