

Tecnología Digital 2

Sistemas de Computación

Licenciatura en Tecnología Digital
Universidad Torcuato Di Tella

Ejercicio 1

(Ejercicio 3.b de la Guía 1)

Considerar un arreglo de punteros a *string*.

Implementar la función `char* superConcatenate(char* v[], int size)`, que toma un arreglo de punteros a *string* y retorna la concatenación de todos los elementos del arreglo.

Ejemplo: `superConcatenate(["Eh_", "oo oo", "+Gol"], 3) → "Eh_oo oo+Gol"`

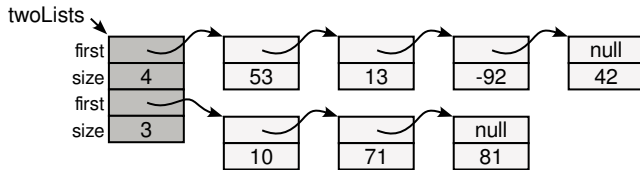
Ejercicio 2

Considerando las estructuras a continuación:

```
struct list {  
    struct node* first;  
    int size;  
}
```

```
struct node {  
    struct node* next;  
    int data;  
}
```

En base a estas estructuras, se construye el siguiente ejemplo, donde el puntero a `twoLists` es un arreglo de dos elementos de tipo `struct list`. La primera lista del ejemplo tiene 4 datos, mientras que la segunda lista tiene 3 datos.



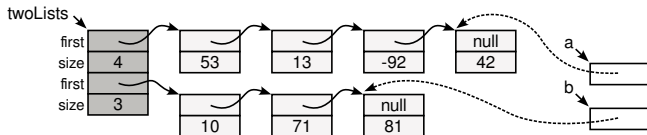
Ejercicio 2

Se pide implementar las siguientes funciones:

❶ `void getLast(struct list* twoLists, struct node** a, struct node** b)`

Toma un puntero a un arreglo de dos listas, recorre ambas listas hasta llegar al último nodo de cada una y escribe en `a` y `b` los punteros a estos últimos nodos. Si las listas pasadas por parámetro no tienen nodos, se debe escribir *null*.

Usando el ejemplo, el resultado de la función pedida es:



❷ ...

Ejercicio 2

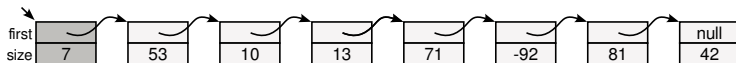
Se pide implementar las siguientes funciones:

1 ...

2 `struct list* getMerge(struct list* twoLists)`

Toma un puntero a un arreglo de dos listas e intercala los elementos de ambas listas en una nueva lista. Comienza por el primer elemento de la primera lista, y luego continua con el primer elemento de la segunda lista, intercalando uno a uno. Las listas pueden ser de distinto tamaño, quedando la lista más larga sin intercalar. La función debe borrar el arreglo de listas pasado por parámetro y retornar una nueva lista.

Usando el ejemplo, el resultado de la función pedida es:



¡Gracias! ;