

TAREA LAB. ESTRUCTURAS DE DATOS CUADERNO DE EJERCICIOS

Aaron Pavez / Alberto Zenteno

Se necesita la creación de un algoritmo en C, que permita imitar la función “split”, presente en otros lenguajes de programación. La función split toma como parámetros: una cadena y un delimitador, retornando una lista con todos elementos encontrados al dividir la cadena por el separador/delimitador.

Ejemplo:

Para la cadena “hola, mi nombre es John Hancock”, al aplicarle un split usando la “,”. Debería retornar “hola” y “mi nombre es John Hancock”.

Para esta misma cadena, si le aplicamos delimitador “ ” (espacio). Debería retornar “hola,” “mi” “nombre” “es” “John” “Hancock”

Si es que no se encuentra el delimitador en la cadena, se devuelve la cadena completa.

El programa deberá recibir como primer parámetro, la frase a separar (entre comillas, para que pueda tomar los espacios”, y el siguiente parámetro el carácter delimitador, como se muestra en la siguiente imagen.

```
[xberto@xberto-pc Escritorio]$ ./untitled "hola don pepito" " "
hola
don
pepito
[xberto@xberto-pc Escritorio]$ ./untitled "hola don pepito" ","
hola don pepito
[xberto@xberto-pc Escritorio]$ ./untitled "hola don,pepito" ","
hola don
pepito
[xberto@xberto-pc Escritorio]$
```

El código debe componerse de una estructura enlazada por punteros, en la cual tenga como variables un arreglo tipo char, donde almacenará cada palabra separada, y un puntero al siguiente elemento que entregue el split.

La función split recibe como parámetros una cadena de caracteres (el texto a “cortar”) y el delimitador, como pueden ver en la línea 10. Y devuelve la cabeza de nuestra estructura enlazada por punteros, la que se irá iterando a través de sus punteros hasta que el puntero “siguiente” sea nulo.

Recuerden revisar la función strtok revisada en clases.

Como verán, entre la línea 14 y la 30 ocurre toda la magia.

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  typedef struct string_arr{
6      char vector[100];
7      struct string_arr *siguiente;
8  }string_arr;
9
10 string_arr *split(char cadena[], char del[]){
11     string_arr *cabeza=NULL;
12     string_arr *aux=NULL;
13     char *ptr = strtok(cadena, del);
14     
15
30
31     return cabeza;
32 }
33
34 int main( int argc, char *argv[] ){
35     if( argc > 3 )
36         printf("Muchos argumentos.\n");
37     else if( argc < 3 )
38         printf("Se esperan dos argumento.\nPrimero la cadena y luego el delimitador");
39     else{
40         string_arr *arreglo = split(argv[1],argv[2]);
41         while(arreglo!=NULL){
42             printf("%s\n", arreglo->vector);
43             arreglo = arreglo->siguiente;
44         }
45     }
46     return 0;
47 }
```

La tarea es individual y puede subirse al Moodle entre el 24 de Abril y 1 de Mayo.
Recordar que tarea que no compile, será evaluada con la nota mínima.