

Piscina C C 12

Índice general

1.	Prei¿½mbulo	
II.	Instrucciones	4
III.	Ejercicio 00 : ft_create_elem	6
IV.	Ejercicio 01 : ft_list_push_front	7
V.	Ejercicio 02 : ft_list_size	8
VI.	Ejercicio 03 : ft_list_last	9
VII.	Ejercicio 04 : ft_list_push_back	10
VIII.	Ejercicio 05 : ft_list_push_strs	11
IX.	Ejercicio 06 : ft_list_clear	12
X.	Ejercicio 07 : ft_list_at	13
XI.	Ejercicio 08 : ft_list_reverse	14
XII.	Ejercicio 09 : ft_list_foreach	15
XIII.	Ejercicio 10 : ft_list_foreach_if	16
XIV.	Ejercicio 11 : ft_list_find	17
XV.	Ejercicio 12 : ft_list_remove_if	18
XVI.	Ejercicio 13 : ft_list_merge	19
XVII.	Ejercicio 14 : ft_list_sort	20
XVIII.	Ejercicio 15 : ft_list_reverse_fun	21
XIX.	Ejercicio 16 : ft_sorted_list_insert	22
XX.	Ejercicio 17 : ft sorted list merge	23

Capítulo I Pre�mbulo

> AVISO DE SPOILER NO LEA LA P�GINA SIGUIENTE

Usted se lo ha buscado.

- En Star Wars, Dark Vador es el padre de Luke Skywalker.
- En Sospechosos habituales, Verbal es Keyser Soze.
- En El club de la lucha, Tyler Durden y el narrador son la misma persona.
- En El sexto sentido, Bruce Willis estï; ½ muerto desde el principio.
- En Los otros, los residentes de la casa son los fantasmas y viceversa.
- En Bambi, muere la madre de Bambi.
- En El bosque, los monstruos son los aldeanos y, en realidad, la historia ocurre en nuestra �poca.
- En Harry Potter, muere Dumbledore.
- En El planeta de los simios, la historia se desarrolla en la Tierra.
- En Juego de tronos, Robb Stark y Joffrey Baratheon mueren en su noche de bodas.
- En Crepi¿%sculo, los vampiros brillan cuando se exponen al sol.
- En Stargate SG-1, Temporada 1, Episodio 18, O'Neill y Carter est�n en la Ant�rtida.
- En El caballero oscuro: la leyenda renace, Miranda Tate es Talia al Ghul.
- En Super Mario Bros, la princesa se encuentra en otro castillo.

Capítulo II

Instrucciones

- \bullet Esta pi
į½gina seri į½ la i į½nica referencia: no se fi į½e de los rumores de pasillo.
- Vuelva a leer bien los enunciados antes de entregar sus ejercicios. Los enunciados pueden cambiar en cualquier momento.
- Tenga cuidado con los permisos de sus archivos y de sus directorios.
- Debe respetar el procedimiento de entrega para todos sus ejercicios.
- Sus compa�eros de piscina se encargar�n de corregir sus ejercicios.
- Adem�s de por sus compa�eros, tambi�n ser� corregido por un programa que se llama la Moulinette.
- La Moulinette es muy estricta a la hora de dar notas. Est� completamente automatizada. Es imposible discutir con ella sobre su nota. Por lo tanto, sea extremadamente riguroso para evitar cualquier sorpresa.
- La Moulinette no tiene una mente muy abierta. No intenta comprender el ci¿½digo que no respeta la Norma. La Moulinette utiliza el programa norminette para comprobar la norma de sus archivos. Entienda entonces que es esti;½pido entregar un ci;½digo que no pase la norminette.
- Los ejercicios han sido ordenados con mucha precisii; ½n del mi; ½s sencillo al mi; ½s complejo. En ningi; ½n caso le prestaremos atencii; ½n ni tendremos en cuenta un ejercicio complejo si no se ha conseguido realizar perfectamente un ejercicio mi; ½s sencillo.
- El uso de una funci�n prohibida se considera una trampa. Cualquier trampa serï;½ sancionada con la nota -42.
- Solamente tendri; ½ que entregar una funcii; ½n main() si le pedimos un programa.
- La Moulinette compila con los flags -Wall -Wextra -Werror y utiliza gcc.
- Si su programa no compila, tendri; ½ 0.
- <u>No debe</u> dejar en su directorio <u>ning�n</u> archivo que no se haya indicado de forma expl�cita en los enunciados de los ejercicios.

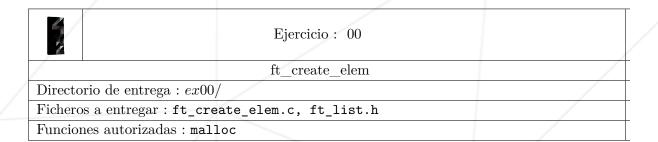
Piscina C

- i¿½Tiene alguna pregunta? Pregunte a su vecino de la derecha. Si no, pruebe con su vecino de la izquierda.
- \bullet Su manual de referencia se llama Google / man / Internet /
- �No olvide participar en el foro Piscina de su Intranet o en el slack de su Piscina!
- Lea detenidamente los ejemplos. Podr�an exigir cosas que no se especifican necesariamente en los enunciados...
- Razone. �Se lo suplico, por Od�n! Maldita sea.
- Para los ejercicios sobre las listas, utilizaremos la estructura siguiente:

- Debe colocar esta estructura en un archivo ft_list.h y entregarlo en cada ejercicio.
- A partir del ejercicio 01 utilizaremos nuestro ft_create_elem, tome las medidas necesarias (podr�a ser interesante tener su prototipo en ft_list.h...).

Capítulo III

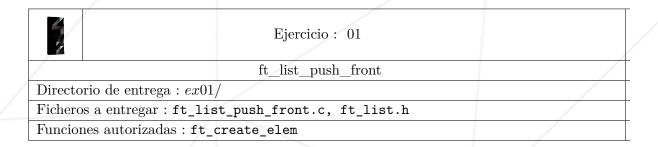
Ejercicio 00: ft_create_elem



- Escriba la funci�n ft_create_elem que cree un elemento nuevo de tipo t_list.
- Tendr� que asignar data al par�metro proporcionado y next a NULL.
- El prototipo de la funcii;½n deberi;½ ser el siguiente:

Capítulo IV

Ejercicio 01 : ft_list_push_front



- Escriba la funci�n ft_list_push_front que a�ada al principio de la lista un elemento nuevo de tipo t_list.
- Tendriį½ que asignar data al pariį½metro proporcionado.
- Actualizarĩ¿½, si es preciso, el puntero al principio de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

void ft_list_push_front(t_list **begin_list, void *data);

Capítulo V

Ejercicio 02 : ft_list_size

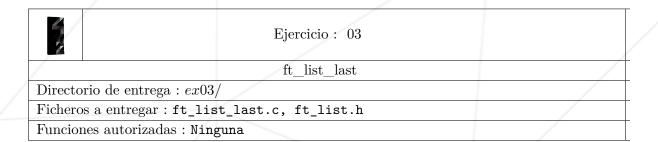
	Ejercicio: 02	
	ft_list_size	/
Directorio de entrega : $ex02/$		
Ficheros a entregar : ft_list_size.c, ft_list.h		
Funciones autorizadas : Ninguna		

- Escriba la funci�n ft_list_size que devuelva el n�mero de elementos de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

int ft_list_size(t_list *begin_list);

Capítulo VI

Ejercicio 03: ft_list_last

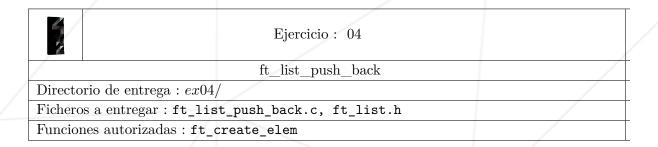


- Escriba la funci�n ft_list_last que devuelva el �ltimo elemento de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

t_list *ft_list_last(t_list *begin_list);

Capítulo VII

Ejercicio 04 : ft_list_push_back

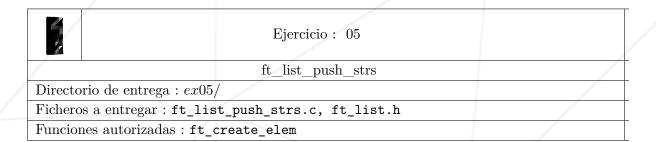


- Escriba la funci�n ft_list_push_back que a�ada al final de la lista un elemento nuevo de tipo t_list.
- Tendri;½ que asignar data al pari;½metro proporcionado.
- Actualizarĩ¿½, si es preciso, el puntero al principio de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

void ft_list_push_back(t_list **begin_list, void *data);

Capítulo VIII

Ejercicio 05 : ft_list_push_strs

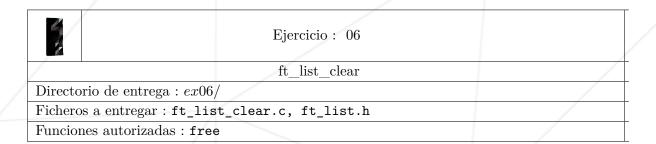


- Escriba la funci�n ft_list_push_strs que cree una lista nueva e introduzca en ella las cadenas de caracteres apuntadas por los elementos de la tabla strs.
- size es el tama�o de strs
- El primer elemento de la tabla se encontrari; ½ al final de la lista.
- Se devolveri; ½ la direccii; ½n del primer elemento de la lista.
- El prototipo de la funcii; ½n deberi; ½ ser el siguiente:

t_list *ft_list_push_strs(int size, char **strs);

Capítulo IX

Ejercicio 06: ft_list_clear



- Escriba la funci�n ft_list_clear que retire y libere todos los elementos de la lista.
- Tambi�n tendr�n que liberarse todos los data usando free_fct.
- El prototipo de la funci�n deber� ser el siguiente:

void ft_list_clear(t_list *begin_list, void (*free_fct)(void *));

Capítulo X

Ejercicio 07: ft_list_at

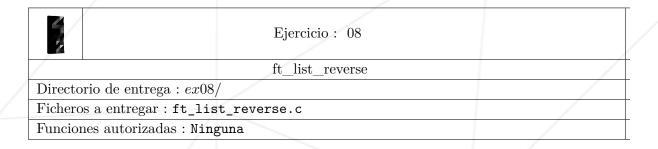
	Ejercicio: 07	
/	ft_list_at	
Directorio de entreg		
Ficheros a entregar : ft_list_at.c, ft_list.h		
Funciones autorizad	as: Ninguna	

- Escriba la funci�n ft_list_at que devuelva el n-�simo elemento de la lista, sabiendo que el primer elemento es el elemento 0.
- En caso de error, devolver� un puntero nulo.
- El prototipo de la funci�n deber� ser el siguiente:

t_list *ft_list_at(t_list *begin_list, unsigned int nbr);

Capítulo XI

Ejercicio 08: ft_list_reverse

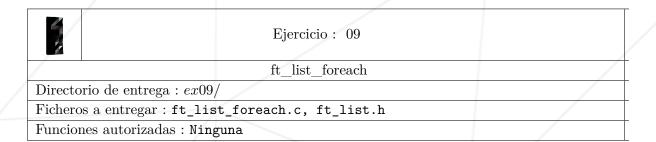


- Escriba la funci�n ft_list_reverse que invierta el orden de los elementos de la lista. Solo se admiten los juegos de punteros.
- Atenci�n, en este ejercicio utilizaremos nuestro propio ft_list.h.
- El prototipo de la funci�n deber� ser el siguiente:

void ft_list_reverse(t_list **begin_list);

Capítulo XII

Ejercicio 09: ft_list_foreach



- Escriba la funci�n ft_list_foreach que aplique una funci�n pasada como par�metro al valor incluido en cada elemento de la lista.
- Se debe aplicar f en el orden de los elementos de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

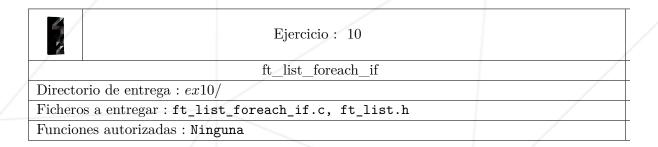
```
void ft_list_foreach(t_list *begin_list, void (*f)(void *));
```

• La funci�n apuntada por f ser� utilizada de la siguiente forma:

(*f)(list_ptr->data);

Capítulo XIII

Ejercicio 10: ft_list_foreach_if



- Escriba la funci�n ft_list_foreach_if que aplique una funci�n pasada como par�metro al valor incluido en algunos elementos de la lista.
- Solo se aplicari; ½ f a los elementos que, al ser pasados como argumento a cmp con data_ref, hagan que cmp devuelva 0.
- Se debe aplicar f en el orden de los elementos de la lista.
- El prototipo de la funci�n deber� ser el siguiente:

```
void ft_list_foreach_if(t_list *begin_list, void (*f)(void *), void
*data_ref, int (*cmp)())
```

• Las funciones apuntadas por f y por cmp seri; ½n utilizadas de la siguiente forma:

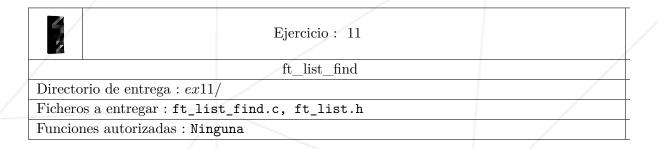
```
(*f)(list_ptr->data);
(*cmp)(list_ptr->data, data_ref);
```



La funciï¿ $\mbox{%} n$ cmp podrï¿ $\mbox{%} a$ ser, por ejemplo, ft_strcmp...

Capítulo XIV

Ejercicio 11: ft_list_find



- Escriba la funci�n ft_list_find que devuelva la direcci�n del primer elemento cuyos datos, cuando se los compare con data_ref usando cmp, hagan que cmp devuelva 0.
- El prototipo de la funci�n deber� ser el siguiente:

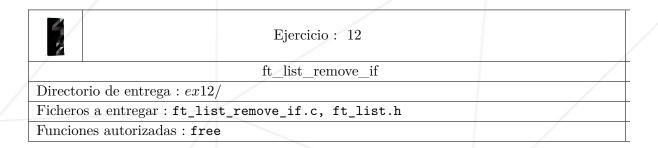
```
t_list *ft_list_find(t_list *begin_list, void *data_ref, int (*cmp)());
```

• La funci�n apuntada por cmp ser� utilizada de la siguiente forma:

(*cmp)(list_ptr->data, data_ref);

Capítulo XV

Ejercicio 12: ft_list_remove_if



- Escriba la funciï; ½n ft_list_remove_if que borre de la lista todos los elementos cuyos datos, cuando se los compare con data_ref usando cmp, hagan que cmp devuelva 0.
- Tambi�n tendr�n que liberarse todos los data de un elemento que se tenga que borrar usando free_fct.
- El prototipo de la funcii;½n deberi;½ ser el siguiente:

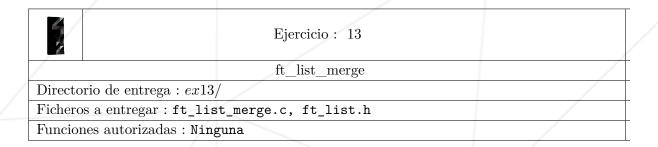
```
void ft_list_remove_if(t_list **begin_list, void *data_ref, int (*cmp)(), void (*free_fct)(void *)
```

• Las funciones apuntadas por free_fct y por cmp ser�n utilizadas de la siguiente forma:

```
(*cmp)(list_ptr->data, data_ref);
(*free_fct)(list_ptr->data);
```

Capítulo XVI

Ejercicio 13: ft_list_merge

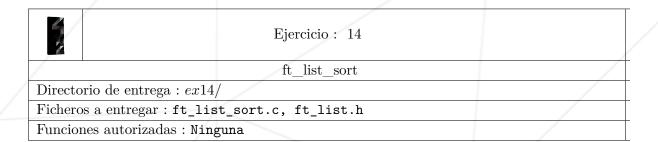


- Escriba la funciï;½n ft_list_merge que coloque los elementos de una lista begin2 al final de otra lista begin1.
- No se permite la creaci�n de elementos.
- El prototipo de la funci�n deber� ser el siguiente:

void ft_list_merge(t_list **begin_list1, t_list *begin_list2);

Capítulo XVII

Ejercicio 14: ft_list_sort



- Escriba la funci�n ft_list_sort que ordene de forma creciente el contenido de la lista, comparando dos elementos mediante una funci�n de comparaci�n de datos de dos elementos.
- El prototipo de la funci�n deber� ser el siguiente:

```
void ft_list_sort(t_list **begin_list, int (*cmp)());
```

• La funci�n apuntada por cmp ser� utilizada de la siguiente forma:

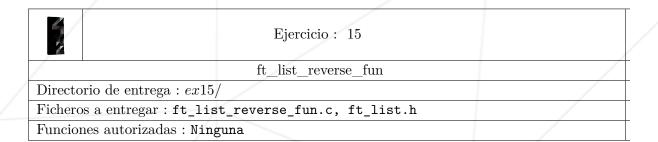
```
(*cmp)(list_ptr->data, other_list_ptr->data);
```



La funci�n cmp podr�a ser, por ejemplo, ft_strcmp.

Capítulo XVIII

Ejercicio 15 : ft__list__reverse__fun

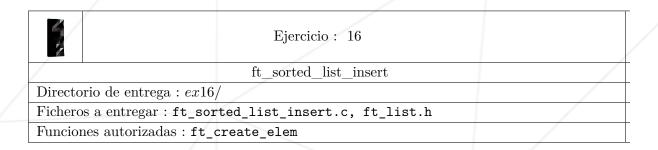


- Escriba la funci�n ft_list_reverse_fun que invierta el orden de los elementos de la lista.
- \bullet El prototipo de la funciï;½
n deberï;½ ser el siguiente:

void ft_list_reverse_fun(t_list *begin_list);

Capítulo XIX

Ejercicio 16: ft_sorted_list_insert



- Escriba la funci�n ft_sorted_list_insert que cree un elemento nuevo y lo inserte en una lista ordenada de tal modo que la lista quede en orden creciente.
- El prototipo de la funci�n deber� ser el siguiente:

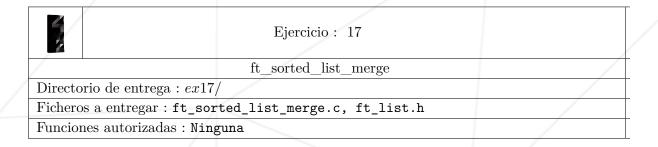
```
void ft_sorted_list_insert(t_list **begin_list, void *data, int (*cmp)());
```

 \bullet La funcii;½n apuntada por ${\tt cmp}$ ser
i;½ utilizada de la siguiente forma:

(*cmp)(list_ptr->data, other_list_ptr->data);

Capítulo XX

Ejercicio 17: ft_sorted_list_merge



- Escriba la funci�n ft_sorted_list_merge que integre los elementos de una lista ordenada begin2 dentro de otra lista ordenada begin1, de tal modo que la lista begin1 quede en orden creciente.
- El prototipo de la funci�n deber� ser el siguiente:

```
void ft_sorted_list_merge(t_list **begin_list1, t_list *begin_list2, int (*cmp)());
```

• La funci�n apuntada por cmp ser� utilizada de la siguiente forma:

(*cmp)(list_ptr->data, other_list_ptr->data);