

Projet C

1.0.0

Generated by Doxygen 1.8.17

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 cell Struct Reference	5
3.2 lcell Struct Reference	6
3.3 list Struct Reference	6
3.4 llist Struct Reference	7
4 File Documentation	9
4.1 src/list.c File Reference	9
4.1.1 Detailed Description	10
4.1.2 Function Documentation	10
4.1.2.1 compare_cells()	10
4.1.2.2 free_list()	10
4.1.2.3 insert()	11
4.1.2.4 load_file()	11
4.1.2.5 make_cell()	11
4.1.2.6 make_cell_from_line()	12
4.1.2.7 new_list()	12
4.1.2.8 pop()	12
4.1.2.9 print_cell()	12
4.1.2.10 print_list()	13
4.1.2.11 push()	13
4.2 src/list.h File Reference	13
4.2.1 Detailed Description	15
4.2.2 Function Documentation	15
4.2.2.1 compare_cells()	15
4.2.2.2 free_list()	16
4.2.2.3 insert()	16
4.2.2.4 load_file()	16
4.2.2.5 make_cell()	16
4.2.2.6 make_cell_from_line()	17
4.2.2.7 new_list()	17
4.2.2.8 pop()	17
4.2.2.9 print_cell()	18
4.2.2.10 print_list()	18
4.2.2.11 push()	18
4.3 src/llist.c File Reference	18
4.3.1 Detailed Description	20

4.3.2 Function Documentation	20
4.3.2.1 compare_lcells()	20
4.3.2.2 free_llist()	20
4.3.2.3 insert_optimized()	21
4.3.2.4 load_file_optimized()	21
4.3.2.5 make_lcell()	21
4.3.2.6 new_llist()	22
4.3.2.7 print_lcell()	22
4.3.2.8 print_llist()	22
4.4 src/llist.h File Reference	23
4.4.1 Detailed Description	24
4.4.2 Function Documentation	24
4.4.2.1 compare_lcells()	25
4.4.2.2 free_llist()	26
4.4.2.3 insert_optimized()	26
4.4.2.4 load_file_optimized()	26
4.4.2.5 make_lcell()	27
4.4.2.6 new_llist()	27
4.4.2.7 print_lcell()	28
4.4.2.8 print_llist()	28
4.5 src/main.c File Reference	28
4.5.1 Detailed Description	29
4.5.2 Function Documentation	30
4.5.2.1 main()	30
Index	33

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

cell	5
lcell	6
list	6
llist	7

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

src/ list.c	This is the list.c file	9
src/ list.h	13
src/ llist.c	This is the llist.c file It implements the 2nd method of the project. In order to optimize the loading of data, we use an index to store more efficiently	18
src/ llist.h	23
src/ main.c	This is the main.c file used to launch the methods for loading files	28

Chapter 3

Data Structure Documentation

3.1 cell Struct Reference

Collaboration diagram for cell:



Data Fields

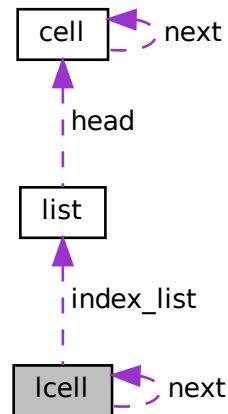
- char * **fname**
- char * **lname**
- char * **zip**
- struct [cell](#) * **next**

The documentation for this struct was generated from the following file:

- src/[list.h](#)

3.2 lcell Struct Reference

Collaboration diagram for lcell:



Data Fields

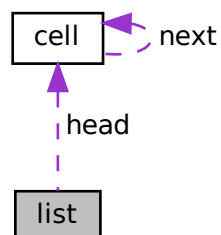
- `char * index`
- `struct list * index_list`
- `struct lcell * next`

The documentation for this struct was generated from the following file:

- `src/list.h`

3.3 list Struct Reference

Collaboration diagram for list:



Data Fields

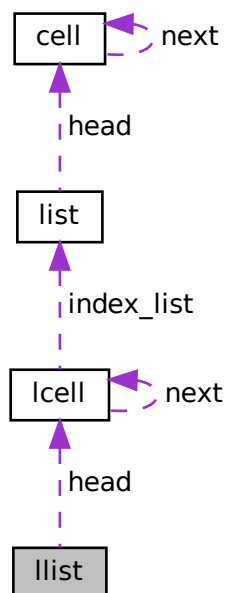
- struct `cell` * `head`

The documentation for this struct was generated from the following file:

- `src/llist.h`

3.4 llist Struct Reference

Collaboration diagram for llist:



Data Fields

- struct `lcell` * `head`

The documentation for this struct was generated from the following file:

- `src/llist.h`

Chapter 4

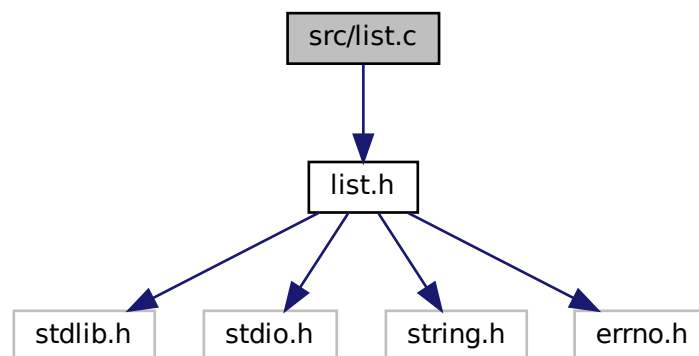
File Documentation

4.1 src/list.c File Reference

This is the [list.c](#) file.

```
#include "list.h"
```

Include dependency graph for list.c:



Functions

- `struct list * new_list ()`
- `void free_list (struct list *lst)`
- `void print_cell (struct cell *c)`
- `void print_list (struct list *lst)`
- `struct cell * make_cell (char *fname, char *lname, char *zip)`
- `void push (struct list *lst, struct cell *c)`
- `void pop (struct list *lst, struct cell *out)`
- `struct cell * make_cell_from_line (char *line)`
- `struct list * load_file (char *file_name)`
- `int compare_cells (struct cell *a, struct cell *b)`
- `void insert (struct list *lst, struct cell *c)`

4.1.1 Detailed Description

This is the [list.c](#) file.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.1.2 Function Documentation

4.1.2.1 `compare_cells()`

```
int compare_cells (
    struct cell * a,
    struct cell * b )
```

Parameters

<i>a</i>	
<i>b</i>	

Returns

int

4.1.2.2 `free_list()`

```
void free_list (
    struct list * lst )
```

Parameters

<i>lst</i>	
------------	--

4.1.2.3 insert()

```
void insert (
    struct list * lst,
    struct cell * c )
```

Parameters

<i>lst</i>	
<i>c</i>	

4.1.2.4 load_file()

```
struct list* load_file (
    char * file_name )
```

Parameters

<i>file_name</i>	
------------------	--

Returns

struct list*

4.1.2.5 make_cell()

```
struct cell* make_cell (
    char * fname,
    char * lname,
    char * zip )
```

Parameters

<i>fname</i>	
<i>lname</i>	
<i>zip</i>	

Returns

struct cell*

4.1.2.6 make_cell_from_line()

```
struct cell* make_cell_from_line (
    char * line )
```

Parameters

<i>line</i>	
-------------	--

Returns

struct cell*

4.1.2.7 new_list()

```
struct list* new_list ( )
```

Returns

struct list*

4.1.2.8 pop()

```
void pop (
    struct list * lst,
    struct cell * out )
```

Parameters

<i>lst</i>	
<i>out</i>	

4.1.2.9 print_cell()

```
void print_cell (
    struct cell * c )
```


Parameters

<i>c</i>	
----------	--

4.1.2.10 print_list()

```
void print_list (
    struct list * lst )
```

Parameters

<i>lst</i>	
------------	--

4.1.2.11 push()

```
void push (
    struct list * lst,
    struct cell * c )
```

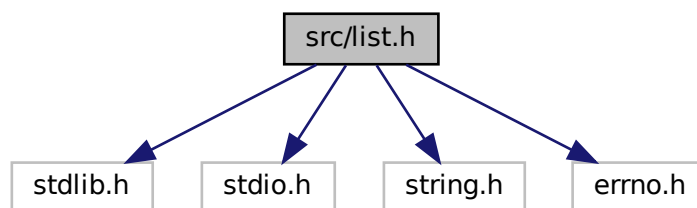
Parameters

<i>lst</i>	
<i>c</i>	

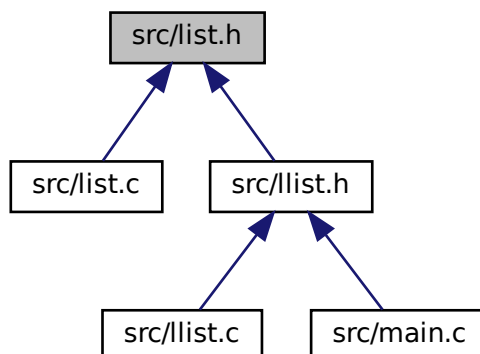
4.2 src/list.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
```

Include dependency graph for list.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [list](#)
- struct [cell](#)

Macros

- #define **NAME_LENGTH** 40
- #define **ZIP_LENGTH** 10

Functions

- struct `list` * `new_list` ()
- void `free_list` (struct `list` *`lst`)
- void `print_cell` (struct `cell` *`c`)
- void `print_list` (struct `list` *`lst`)
- struct `cell` * `make_cell` (char *`fname`, char *`lname`, char *`zip`)
- struct `cell` * `make_cell_from_line` (char *`line`)
- void `push` (struct `list` *`lst`, struct `cell` *`c`)
- void `pop` (struct `list` *`lst`, struct `cell` *`out`)
- int `compare_cells` (struct `cell` *`a`, struct `cell` *`b`)
- void `insert` (struct `list` *`lst`, struct `cell` *`c`)
- struct `list` * `load_file` (char *`file_name`)

4.2.1 Detailed Description

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.2.2 Function Documentation

4.2.2.1 `compare_cells()`

```
int compare_cells (  
    struct cell * a,  
    struct cell * b )
```

Parameters

<i>a</i>	
<i>b</i>	

Returns

int

4.2.2.2 free_list()

```
void free_list (
    struct list * lst )
```

Parameters

<i>lst</i>	
------------	--

4.2.2.3 insert()

```
void insert (
    struct list * lst,
    struct cell * c )
```

Parameters

<i>lst</i>	
<i>c</i>	

4.2.2.4 load_file()

```
struct list* load_file (
    char * file_name )
```

Parameters

<i>file_name</i>	
------------------	--

Returns

struct list*

4.2.2.5 make_cell()

```
struct cell* make_cell (
    char * fname,
```

```
char * lname,  
char * zip )
```

Parameters

<i>fname</i>	
<i>lname</i>	
<i>zip</i>	

Returns

struct cell*

4.2.2.6 make_cell_from_line()

```
struct cell* make_cell_from_line (  
    char * line )
```

Parameters

<i>line</i>	
-------------	--

Returns

struct cell*

4.2.2.7 new_list()

```
struct list* new_list ( )
```

Returns

struct list*

4.2.2.8 pop()

```
void pop (  
    struct list * lst,  
    struct cell * out )
```

Parameters

<i>lst</i>	
<i>out</i>	

4.2.2.9 print_cell()

```
void print_cell (
    struct cell * c )
```

Parameters

<i>c</i>	
----------	--

4.2.2.10 print_list()

```
void print_list (
    struct list * lst )
```

Parameters

<i>lst</i>	
------------	--

4.2.2.11 push()

```
void push (
    struct list * lst,
    struct cell * c )
```

Parameters

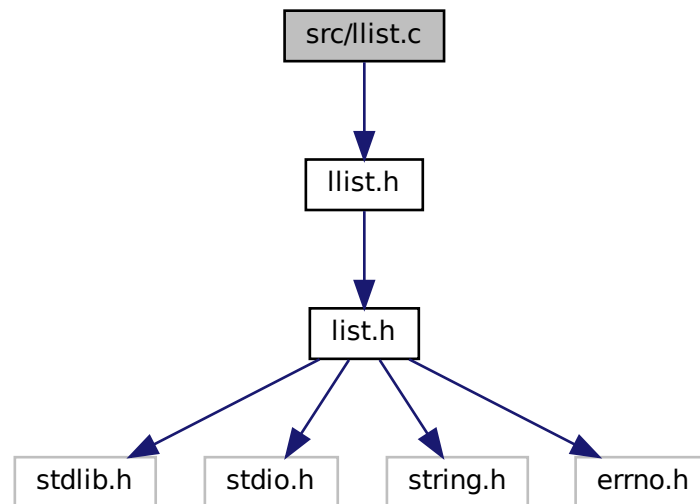
<i>lst</i>	
<i>c</i>	

4.3 src/llist.c File Reference

This is the `llist.c` file It implements the 2nd method of the project. In order to optimize the loading of data, we use an index to store more efficiently.

```
#include "llist.h"
```

Include dependency graph for llist.c:



Functions

- `struct llist * new_llist ()`
new_llist allocates memory for a new llist and returns its pointer.
- `void free_llist (struct llist *llst)`
free_llist frees the llist by calling free_lcells and then frees the llist pointer.
- `void print_lcell (struct lcell *lcell)`
Prints the lcell content using a list format :
Example of printing :

```
[A,
{
[Firstname,Aname,1234],
}
,B]
```

Therefore, it shows the content of the current index letter, the index_list and the next index letter.
- `void print_llist (struct llist *llst)`
Prints the llist using a format like :

```
{{
"prints the lcells"
}}.
```
- `struct lcell * make_lcell (struct list *index_list, struct cell *c)`
make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)
- `int compare_lcells (struct lcell *lcellule, struct cell *c)`
Compare cells to the index to insert it to the right place.
- `void insert_optimized (struct llist *llst, struct cell *c)`
Add the cell to the llist respecting alphabetical order of names/lname and inserting into the right index to optimize the llist.
- `struct llist * load_file_optimized (char *file_name)`
Load the contents of a file in a sorted list.

4.3.1 Detailed Description

This is the `llist.c` file. It implements the 2nd method of the project. In order to optimize the loading of data, we use an index to store more efficiently.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.3.2 Function Documentation

4.3.2.1 `compare_lcells()`

```
int compare_lcells (
    struct lcell * lcellule,
    struct cell * c )
```

Compare cells to the index to insert it to the right place.

Parameters

<i>lcellule</i>	
<i>c</i>	

Returns

int

4.3.2.2 `free_llist()`

```
void free_llist (
    struct llist * llst )
```

`free_llist` frees the `llist` by calling `free_lcells` and then frees the `llist` pointer.

Parameters

<i>l1st</i>	
-------------	--

4.3.2.3 insert_optimized()

```
void insert_optimized (
    struct l1st * l1st,
    struct cell * c )
```

Add the cell to the l1st respecting alphabetical order of names/lnames and inserting into the right index to optimize the l1st.

Parameters

<i>l1st</i>	
<i>c</i>	

4.3.2.4 load_file_optimized()

```
struct l1st* load_file_optimized (
    char * file_name )
```

Load the contents of a file in a sorted list.

Parameters

<i>file_name</i>	
------------------	--

Returns

struct l1st*

4.3.2.5 make_lcell()

```
struct lcell* make_lcell (
    struct list * index_list,
    struct cell * c )
```

make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)

Parameters

<i>index_list</i>	
<i>c</i>	

Returns

struct lcell*

4.3.2.6 new_llist()

```
struct llist* new_llist ( )
```

new_llist allocates memory for a new llist and returns its pointer.

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

4.3.2.7 print_lcell()

```
void print_lcell (
    struct lcell * lcell )
```

Prints the lcell content using a list format :

Example of printing :

```
[A,
{
[Firstname,Aname,1234],
}
,B]
```

Therefore, it shows the content of the current index letter, the index_list and the next index letter.

Parameters

<i>lcell</i>	
--------------	--

4.3.2.8 print_llist()

```
void print_llist (
    struct llist * llist )
```

Prints the l1ist using a format like :

```
{  
"prints the lcells"  
}
```

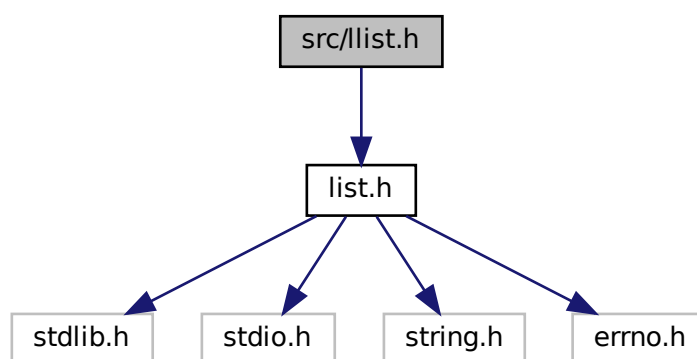
Parameters

<i>l1ist</i>	
--------------	--

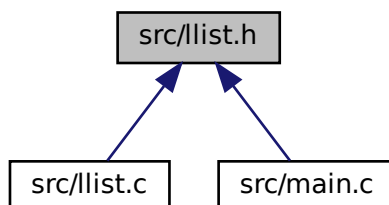
4.4 src/l1ist.h File Reference

```
#include "list.h"
```

Include dependency graph for l1ist.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [l1ist](#)
- struct [lcell](#)

Functions

- struct `llist` * `new_llist` ()
Allocate memory for a llist structure and return the pointer.
- void `free_llist` (struct `llist` *`llst`)
free_llist frees the llist by calling free_lcells and then frees the llist pointer.
- void `print_lcell` (struct `lcell` *`lcell`)
Prints the lcell content using a list format :
Example of printing :

```
[A,
{
[Firstname,Aname,1234],
}
,B]
```

Therefore, it shows the content of the current index letter, the index_list and the next index letter.
- void `print_llist` (struct `llist` *`llst`)
Prints the llist using a format like :

```
{{
"prints the lcells"
}}.
```
- struct `lcell` * `make_lcell` (struct `list` *`index_list`, struct `cell` *`c`)
make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)
- int `compare_lcells` (struct `lcell` *`lcellule`, struct `cell` *`c`)
Compare cells to the index to insert it to the right place.
- void `insert_optimized` (struct `llist` *`llst`, struct `cell` *`c`)
Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.
- struct `llist` * `load_file_optimized` (char *`file_name`)
Load the contents of a file in a sorted list.

4.4.1 Detailed Description

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.4.2 Function Documentation

4.4.2.1 compare_lcells()

```
int compare_lcells (
    struct lcell * lcellule,
    struct cell * c )
```

Compare cells to the index to insert it to the right place.

Parameters

<i>lcellule</i>	
<i>c</i>	

Returns

int

4.4.2.2 free_llist()

```
void free_llist (
    struct llist * llst )
```

free_llist frees the llist by calling free_lcells and then frees the llist pointer.

Parameters

<i>llst</i>	
-------------	--

4.4.2.3 insert_optimized()

```
void insert_optimized (
    struct llist * llst,
    struct cell * c )
```

Add the cell to the llist respecting alphabetical order of names/lnames and inserting into the right index to optimize the llist.

Parameters

<i>llst</i>	
<i>c</i>	

4.4.2.4 load_file_optimized()

```
struct llist* load_file_optimized (
    char * file_name )
```

Load the contents of a file in a sorted list.

Parameters

<i>file_name</i>	
------------------	--

Returns

struct llist*

4.4.2.5 make_lcell()

```
struct lcell* make_lcell (
    struct list * index_list,
    struct cell * c )
```

make_lcell creates a lcell by using the first character of the c->lname and inserts the cellule c into the index_list It also allocates the memory for the lcellule and its core components. (index, index_list)

Parameters

<i>index_list</i>	
<i>c</i>	

Returns

struct lcell*

4.4.2.6 new_llist()

```
struct llist* new_llist ( )
```

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

Allocate memory for a llist structure and return the pointer.

Returns

struct llist*

4.4.2.7 print_lcell()

```
void print_lcell (
    struct lcell * lcell )
```

Prints the lcell content using a list format :

Example of printing :

```
[A,
{
[Firstname,Aname,1234],
}
,B]
```

Therefore, it shows the content of the current index letter, the index_list and the next index letter.

Parameters

<i>lcell</i>	
--------------	--

4.4.2.8 print_llist()

```
void print_llist (
    struct llist * llist )
```

Prints the llist using a format like :

```
{{
"prints the lcells"
}}.
```

Parameters

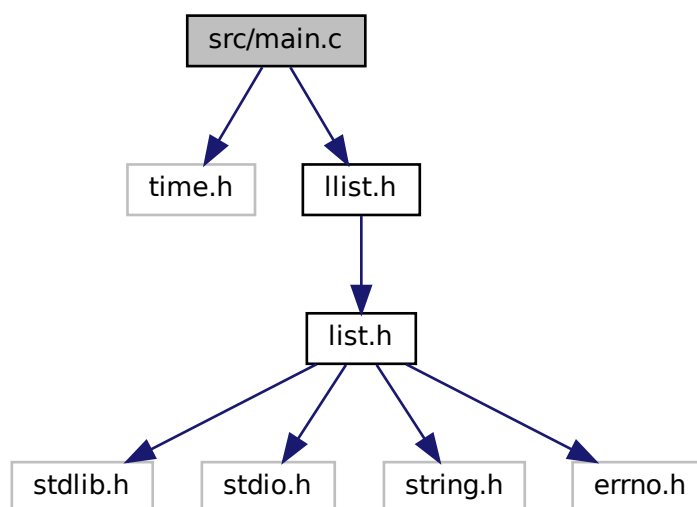
<i>llst</i>	
-------------	--

4.5 src/main.c File Reference

This is the [main.c](#) file used to launch the methods for loading files.

```
#include <time.h>
#include "llist.h"
```


Include dependency graph for main.c:



Functions

- int [main](#) (int argc, char *argv[])

4.5.1 Detailed Description

This is the [main.c](#) file used to launch the methods for loading files.

Author

Mathis URIEN (LBF38)

Version

1.0.0

Date

2022-10-19

Copyright

Copyright (c) 2022

4.5.2 Function Documentation

4.5.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

int

Index

- cell, [5](#)
- compare_cells
 - list.c, [10](#)
 - list.h, [15](#)
- compare_lcells
 - llist.c, [20](#)
 - llist.h, [24](#)
- free_list
 - list.c, [10](#)
 - list.h, [16](#)
- free_llist
 - llist.c, [20](#)
 - llist.h, [26](#)
- insert
 - list.c, [11](#)
 - list.h, [16](#)
- insert_optimized
 - llist.c, [21](#)
 - llist.h, [26](#)
- lcell, [6](#)
- list, [6](#)
- list.c
 - compare_cells, [10](#)
 - free_list, [10](#)
 - insert, [11](#)
 - load_file, [11](#)
 - make_cell, [11](#)
 - make_cell_from_line, [12](#)
 - new_list, [12](#)
 - pop, [12](#)
 - print_cell, [12](#)
 - print_list, [13](#)
 - push, [13](#)
- list.h
 - compare_cells, [15](#)
 - free_list, [16](#)
 - insert, [16](#)
 - load_file, [16](#)
 - make_cell, [16](#)
 - make_cell_from_line, [17](#)
 - new_list, [17](#)
 - pop, [17](#)
 - print_cell, [18](#)
 - print_list, [18](#)
 - push, [18](#)
- llist, [7](#)
- llist.c
 - compare_lcells, [20](#)
 - free_llist, [20](#)
 - insert_optimized, [21](#)
 - load_file_optimized, [21](#)
 - make_lcell, [21](#)
 - new_llist, [22](#)
 - print_lcell, [22](#)
 - print_llist, [22](#)
- llist.h
 - compare_lcells, [24](#)
 - free_llist, [26](#)
 - insert_optimized, [26](#)
 - load_file_optimized, [26](#)
 - make_lcell, [27](#)
 - new_llist, [27](#)
 - print_lcell, [27](#)
 - print_llist, [28](#)
- load_file
 - list.c, [11](#)
 - list.h, [16](#)
- load_file_optimized
 - llist.c, [21](#)
 - llist.h, [26](#)
- main
 - main.c, [30](#)
- main.c
 - main, [30](#)
- make_cell
 - list.c, [11](#)
 - list.h, [16](#)
- make_cell_from_line
 - list.c, [12](#)
 - list.h, [17](#)
- make_lcell
 - llist.c, [21](#)
 - llist.h, [27](#)
- new_list
 - list.c, [12](#)
 - list.h, [17](#)
- new_llist
 - llist.c, [22](#)
 - llist.h, [27](#)
- pop
 - list.c, [12](#)
 - list.h, [17](#)
- print_cell
 - list.c, [12](#)

- list.h, [18](#)
- print_lcell
 - llist.c, [22](#)
 - llist.h, [27](#)
- print_list
 - list.c, [13](#)
 - list.h, [18](#)
- print_llist
 - llist.c, [22](#)
 - llist.h, [28](#)
- push
 - list.c, [13](#)
 - list.h, [18](#)
- src/list.c, [9](#)
- src/list.h, [13](#)
- src/llist.c, [18](#)
- src/llist.h, [23](#)
- src/main.c, [28](#)