

List.c File Reference

```
#include "List.h"
```

Functions

```
List * newList (compFun cmpfn, prFun dspfn)
```

```
void delList (List *list)
```

```
status nthInList (List *list, int nidex, void **elem)
```

```
status addListAt (List *list, int index, void *elem)
```

```
status remFromListAt (List *list, int index, void **out)
```

```
status remFromList (List *list, void *elem)
```

```
status displayList (List *list)
```

```
void forEach (List *list, void(*pFun)(void *))
```

```
int lengthList (List *list)
```

```
status addList (List *list, void *elem)
```

```
Node * isInList (List *list, void *elem)
```

Function Documentation

```
status addList ( List * ,  
                void *  
                )
```

add given element to given list according to comparison function.

Parameters

```
status addListAt ( List * ,  
                  int ,  
                  void *  
                )
```

Insert element at a given position in the list.

Parameters

```
void delList ( List * )
```

destroy the list by deallocating used memory.

Parameters

```
status displayList ( List * )
```

display list the elements according to display function provided during list creation

Parameters

```
void forEach ( List * ,  
              void (*)(void *)  
            )
```

sequentially call given function with each element of given list ($O(N \times F)$).

Parameters

```
Node* isInList ( List * ,  
                void *  
                )
```

tests whether the list contains given element.

Parameters

```
int lengthList ( List * )
```

compute and return the number of elements in given list.

Parameters

```
List* newList ( compFun ,  
               prFun  
               )
```

Empty **List** creation by dynamic memory allocation.

Parameters

```
status nthInList ( List * ,  
                 int      ,  
                 void **  
                 )
```

get the Nth element of the list.

Parameters

```
status remFromList ( List * ,  
                    void *  
                    )
```

remove given element from given list. user should provide a comparison function during list creation.

Parameters

```
status remFromListAt ( List * ,  
                     int ,  
                     void **  
                     )
```

remove the element located at a given position in list.

Parameters