

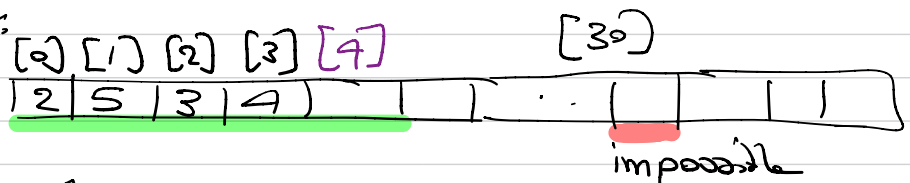
type elem

typedef double elem

- initList
- getNumElem
- isListFull
- isListEmpty

- insertElem
- displayList

Exercice :



numElem = 4

insertion n'est effectuée à une position "pas"

pos

0

au début

1

2

3

4

à la fin ← numElem

-1

à la fin

pos : -1 ou [0 .. numElem]

insert pos = 0 :

[0] → [numElem - 1] d'un cran à droite

insert pos = 2 :

[2] → [numElem - 1] d'un cran à droite

⚠ la liste ne doit pas être pleine.

LISTE TABLEAU → Gestion de types multiples

```
typedef double elem;
```

→ utilisé dans { list.c
 .h

```
typedef double elem;      typedef struct {  
                           int32_t row;  
                           int32_t col;  
                           } elem;
```


→ listType.h

```
#pragma once  
#define ELEM_DOUBLE  
// #define ELEM_POINTED ← define exclusif  
  
#ifdef ELEM_DOUBLE  
    typedef double elem;  
#endif  
  
#ifdef ELEM_POINTED  
    typedef struct {  
        int32_t row;  
        int32_t col;  
    } elem;  
#endif
```



list.h → • inclure "listType.h"
 • Supprimer typedef double elem;

COMPILE CONDITIONAL, via make


ListType.h



```
// #define ELEM_DOUBLE  
// #define ELEM_POINT2D  
// #define ELEM_INT
```



```
make CFLAGS = -D=ELEM_INT
```



```
make CFLAGS = -D=ELEM_POINT2D
```

CFLAGS = -D = ELEM_INT

↳ #define ELEM_INT.

► Recherche de la position d'un élément

int32_t SearchElem (sList *l, elem e);

↑
Position de l'élément e dans la liste l

si -1 → élément non trouvé
sinon [0 ... l->numElem[

Algorithme

► boucle pour parcourir les éléments de la liste

[0] [1] [2] [3] [4] l->numElem = 5

4	12	6	8	1				...	
---	----	---	---	---	--	--	--	-----	--

elem e = 6

indice de boucle. k = 0, 1, 2, 3, 4

condition 1 de sortie de boucle: $k \geq \text{numElem}$

condition 2 de sortie de boucle: on a trouvé!

while ($k < l \rightarrow \text{numElem}$ & on a pas trouvé)

[si e = élément en position k. alors :

[- mémoriser la position k

[- on a trouvé

incrémenter k.

tar; #define ELEM_INT

Essai avec la recherche d'un point 2D. (10-20).
Pour le prog en compléte.
26-27-2020

16⁴28