

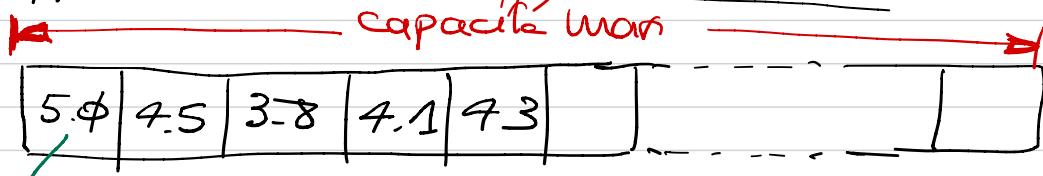
INFO2 TIN - LISTES

17.5.2020

- Liste de fichiers
 - Liste de WayPoints
 - Liste de notes
 - Liste de produits (code barres)
- chaînes de caract.
flong, lat, alt p
double
chaine

(!) Dans une liste, tous les éléments ont le même type

* Support pour stocker les données: TABLEAU



type de la donnée

nbre d'éléments actuellement dans la liste.

double 1) Type de donnée pris dans la liste

define 2) Fixer la capacité de la liste

struct 3) Tableau

 4) Indicateur pour le nombre d'éléments présents actuellement dans la liste

TD 20200514

TD 20200514.c / .h

list.c / .h → création des points
1) 2) et 3)

15'20

```

// preprocessor symbols
#define MAX_LIST_SIZE(100)

// preprocessor macros

// enumerated types

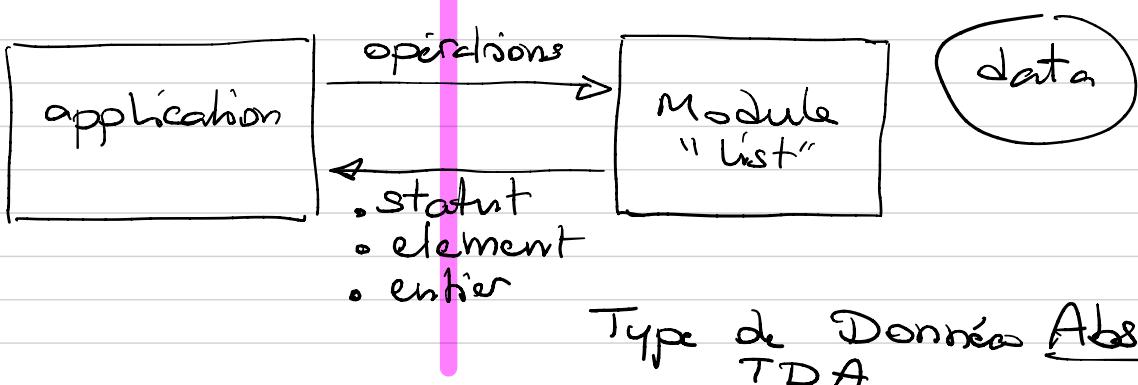
// structured types
typedef double elem; // create the 'elem' type, based on the 'double' type

typedef struct {
    uint32_t numElements;
    elem element[MAX_LIST_SIZE];
} sList;

```

- verbe = action
- ❶ Opérations que l'on veut effectuer sur la liste?
 - ❷
 - initialiser la liste
 - ajouter un élément
 - supprimer un élément
 - rechercher la position d'un élément
 - accéder à un élément
 - modifier un élément
 - recevoir le nombre d'éléments dans la liste
 - savoir si la liste est pleine
 -
 - ❸
 - ❹

TDA



Type de Données Abstrait
TDA

► Initialisation initList

$\ell \rightarrow \text{numElements} \leftarrow \emptyset$

{ main
sList ℓ ;

 init(& ℓ);

► Récupérer le nombre d'éléments uint32_t getNumElem

renvoyer $\ell \rightarrow \text{numElements}$

► Savoir si la liste est pleine

bool
isListFull

isListEmpty

 " " " "

IS455

→ ajout d'un élément

[0] [1] [2] [3] [4]

5	4.5	3	2.1	7		
---	-----	---	-----	---	--	--

① Element 6φ

② Position

numElem = 5

(I)

Position : au début

[0] [1] [2] [3] [4] [5]

6φ	5	4.5	3	2.1	7		
----	---	-----	---	-----	---	--	--

1) déplacement de tous les éléments →
d'une position à droite.

2) mettre l'élément à la posn [φ]

(II)

Position : à la fin

[0] [1] [2] [3] [4] [5]

5	4.5	3	2.1	7	6φ	
---	-----	---	-----	---	----	--

1) mettre l'élément à la posn [numElem]

(III)

Position "au milieu" entre φ et numElem - 1

[0] (1) (2) (3) (4) (5)

5	4.5	6φ	3	2.1	7	
---	-----	----	---	-----	---	--

pos = 2



1) Déplacer des éléments qui sont à partir
de la pos d'1 cran à droite.

2) écrire le l'élément à la posn [pos]

void insertElem(slist *l, elem e, int32_t pos);

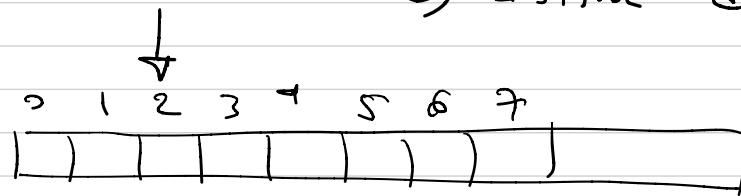
pos: φ pour "à la fin"
-1 pour "à la fin"

insertElem (l, e , pos)

Si $pos == -1 \rightarrow pos = l \rightarrow numElem$

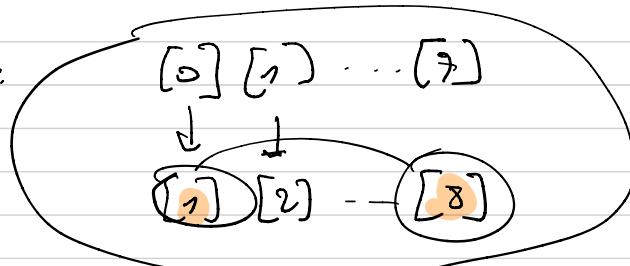
Si il y a 1) déplacent

2) insertion de l'élément



{

insert pos \neq :

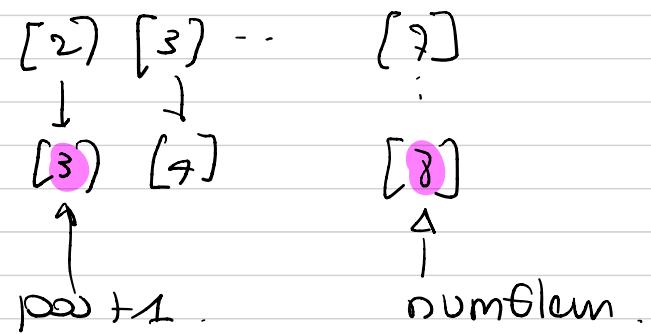


déplacent vers la droite : la boucle doit iterer de la droite vers la gauche.

for(k = numElem ; k >= pos+1 , k--)

{

insert pos 2.



• Terminez la fonction insertElem .

• Ajouter une fonction displayList

↳ affiche le nombre d'élément
↳ les éléments

par le 19/11/2022