



Fonctions pour la gestion de la liste.

- quelle est la taille de la liste ?
- la liste est-elle vide ? plein ?
- vider la liste
- insérer un élément → position: début / fin / pas. particulier
- accéder à un élément → position, valeur
- rechercher un élément → valeur
- supprimer un élément → position, valeur

stratégie si la liste
est pleine ?
→ erreur
→ agrandir la liste

Liste tableau à taille fixe

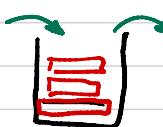


Liste tableau à taille dynamique.

- au départ elle a une taille N (malloc)
- quand est pleine → agrandissement du tableau (realloc) Facteur d'agrandissement

Pile

empile



dépile

Pile "LIFO"

premier élément empilé sera le dernier dépilable

$$\text{First In} = \text{Last In}$$

$$\text{Last Out} = \text{First Out}$$

Queue d'éléments

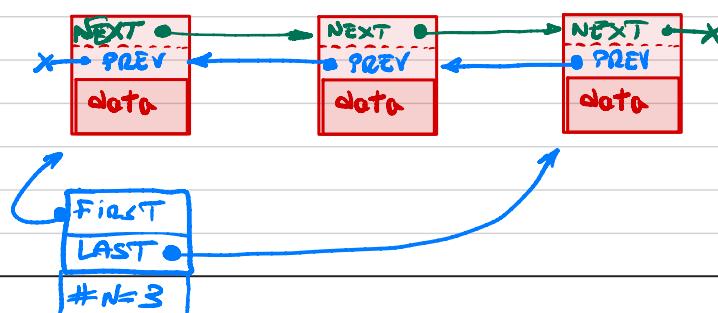
"FIFO"

ajout → {  } → retirer

First In

First Out

Liste chaînée:



Liste Tableau

type d'élément : double
capacité max de la liste : 10

TD 2020 S2

TD 2020 S2.c

← main / manipulation List

error.c / .h
list.c / .h

← cod, error + display

list.h :

typedef double Element; // Element: c'est le type unique qu'on va manipuler dans list.c

// structure spécifique pour la liste:

- nombre d'éléments présent dans la liste
- tableau d'éléments de taille CAPACITY

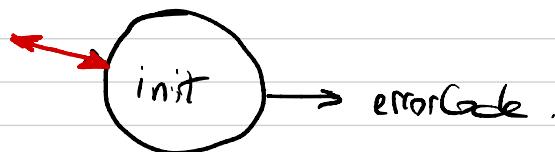
mettre en place les différents éléments

sList

16^h 25

① Fonction d'initialisation de la liste.
numElem ← 0

Liste



② Fonction list vide :

Liste



③ Fonction list Pleine

Liste



16^h 37

prototypes → .h
implémentation → .c

// list.h

#pragma once

#include <stdint.h>
#include <stdbool.h>
#include "error.h"

#define LIST_CAPACITY 10 // list size

typedef double Element; // element in list

typedef struct {
 uint32_t numElem;
 Element e[LIST_CAPACITY];
} sList, *pList;

```

eErrorCode initList(pList l);
bool isEmpty(sList l);
bool isFull(sList l);
  
```

⑦ Fonction pour obtenir le nbr d'élément (getNbElem)

Liste \rightarrow getNbElem \rightarrow nbr d'élément

Insertion d'élément

(AVANT) \rightarrow

[$-$] [1] [2] [3]

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

4 élém.

N

Insertion au début $p=0$ [N] [1] [2] [3] [4]

[0]

5 élém.

Insertion à la fin $p=4$ [1] [2] [3] [4] [N]

[4]

5 élém.

Insertion à la position

$p=2$ em poser

$p=2$ [1] [2] [N] [3] [4]

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

5 élém.

Algorithm pour l'insertion à la position p :

Déplacer d'une position vers la droite tous les éléments suivis à partir de l'index p .

ex: insert au début: $p=\emptyset$
à la fin: $p=$ nbr. élém.
en p :

Déplacement des éléments vers la droite, on fait
se déplacé à droite, à gauche.

N=4

P=2

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

[1] [2] [3] [4] [$+$]

$[n-1] \rightarrow [n]$

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

$[n-2] \rightarrow [n-1]$

$\dots n-k \dots$

[1] [2] [N] [3] [4]

on stoppe si $n-k=p$

en vert: éléments déplacés
en rouge: l'élément ajouté
en bleu: les éléments inchangés

17/10

Liste à
 modifier :
 position d'insertion
 (-1 : à la fin)
 valeur de l'élément
 à insérer

```

eErrorCode insertElemAt(pList l, int32_t pos, Element e) {
  eErrorCode returnCode = E_NO_ERROR;
  int32_t k = 0;
  int32_t n = 0;      test si liste valide

  if (l != NULL)     vérif si liste pas pleine
  {
    if (!isListFull(*l)) {
      n = l->numElem;
      if (pos == -1) [calcule la position d'insertion]
      pos = n; [si insertion à la fin]
      for (k = n - 1; k >= pos; k--) [boucle de
      {                                déplacement
        l->e[k + 1] = l->e[k];       des éléments situés
      }                                à droite de p
      l->e[pos] = e;                 (de droite à gauche)
      l->numElem++;                déplace 1 élément vers
      else {                         insertion du nouvel élément
        M.A.J. du nbr d'élément
        returnCode = E_LIST_FULL;    si liste pleine
      }
    }
    else {
      returnCode = E_BAD_LIST;     si liste non valide.
    }
  }
  return returnCode;
}
  
```

b-droite

Pour Jeudi: • fonction de suppression d'un élément
27.IV à une position "p"
 • Lecture d'un élément à une position "p"
 • Recherche du premier du tableau par sa valeur.

Semaine 13				
Date	Horaire	Cours	Enseignant(s)	Salle(s) (* : A distance)
Lundi 24.05.2021	Férié			
Jeudi 27.05.2021	13:15 - 14:45 (2 périodes)	Info2Mi-B-L1	PBR, MEC	A05b
Jeudi 27.05.2021	16:35 - 18:05 (2 périodes)	Info2Mi-B-C2	PBR	* G04