

# Chapitre 6. Pile/File

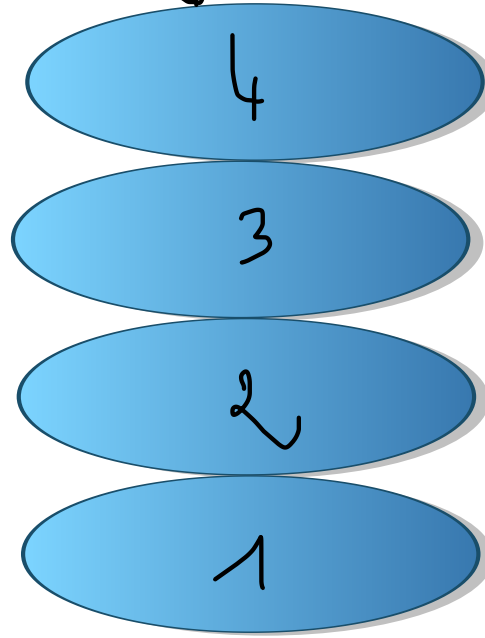
## 1. La structure Pile

Exemple :

Règle: LIFO: last in first out  
Dernier entré premier sorti

depiler = retirer

insérer = empiler



Pile d'assietes

Applications qui utilisent une pile:

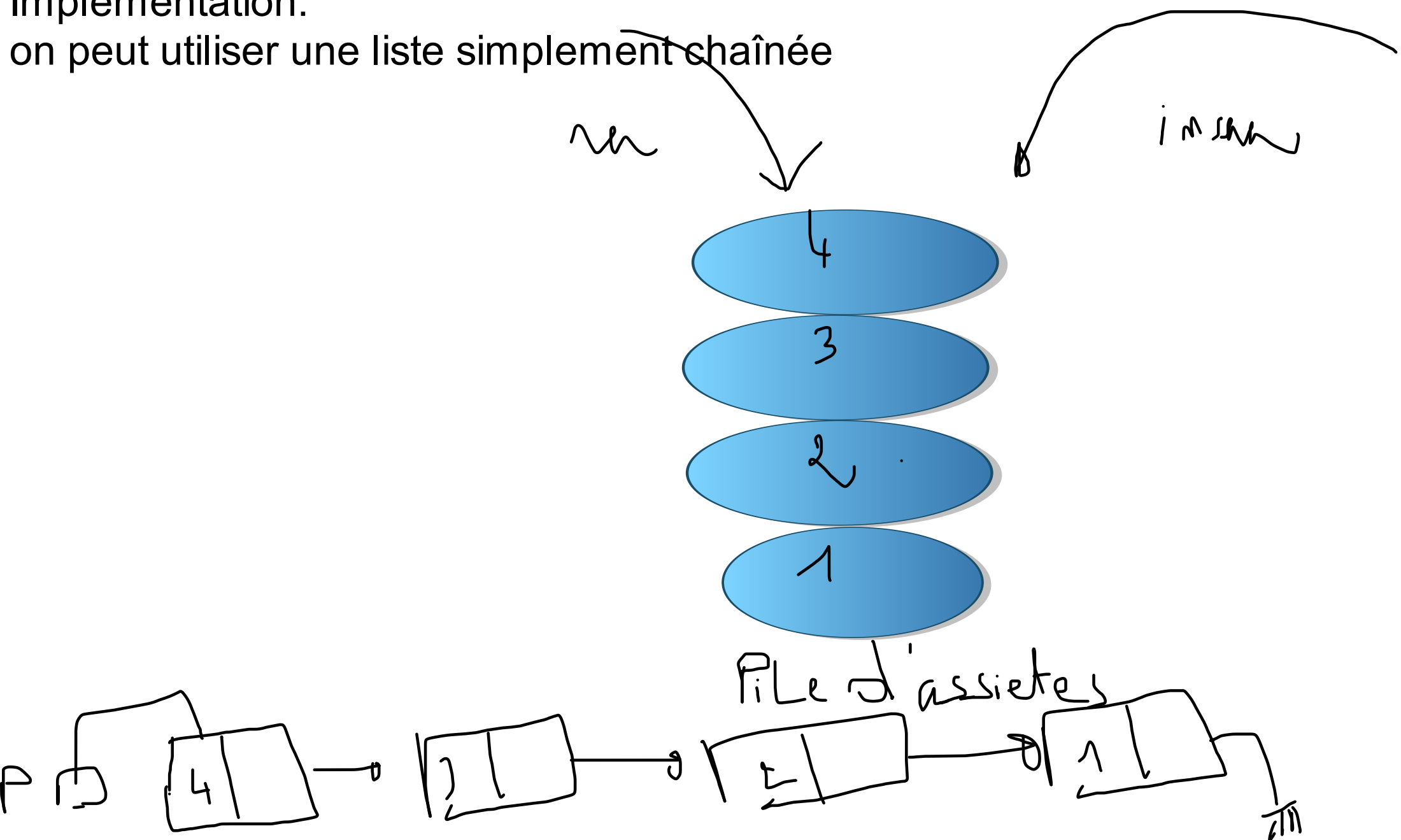
## 1. Evaluation d'une expression arithmétique

$$5 * (4 + 3) = 35^*$$

## 2. Vérification de parenthésage

$$\left( (a + b) * \left( \frac{1}{2} + c \right) \right)$$

Implémentation:  
on peut utiliser une liste simplement chaînée



## Les primitives

1. Pile Vide ( )

2. EstVide ( )

3. Empiler ( )

4. Depiler ( )

5. sommetPile ( )

```
//Définir la structure liste
typedef struct cellule{
    int info;
    struct cellule *suiv;
}liste;
//Définir la structure Pile
typedef liste * Pile
///Les primitives
Pile PileVide()
{
}

/////
int EstPileVide(Pile P)
/////
Pile Empiler(Pile P,int x)
/////
Pile Depiler(Pile P)
/////
int SommetPile(Pile P)
```

```
le sommet de la pile est 2  
le sommet de la pile est 3
```

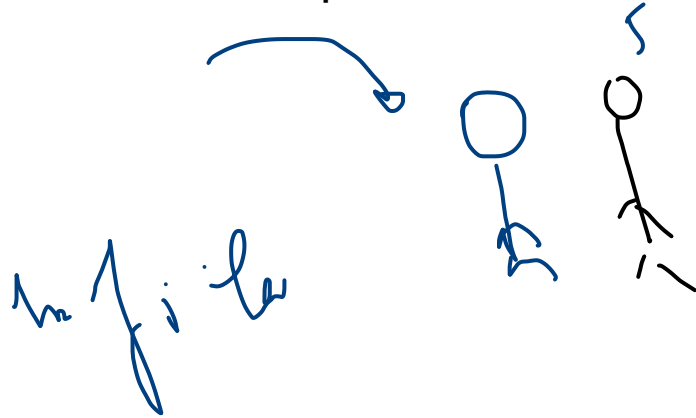
```
main()
{
    Pile P;
    P=PileVide();
    P=Empiler(P,2);
    printf("le sommet de la pile est %d\n",SommetPile(P));
    P=Empiler(P,3);
    printf("le sommet de la pile est %d\n",SommetPile(P));
    P=Empiler(P,1);
    printf("le sommet de la pile est %d\n",SommetPile(P));
    P=Depiler(P);
    printf("le sommet de la pile est %d\n",SommetPile(P));
    P=Depiler(P);
    printf("le sommet de la pile est %d\n",SommetPile(P));
    P=Depiler(P);
    printf("le sommet de la pile est %d\n",SommetPile(P));
}
```

```
le sommet de la pile est 2
le sommet de la pile est 3
le sommet de la pile est 1
le sommet de la pile est 3
le sommet de la pile est 2
la pile est vide
-----
```

## 2. La structure File

règle: FIFO (first in first out)

Premier arrivé premier sorti



def file = supprimer le premier



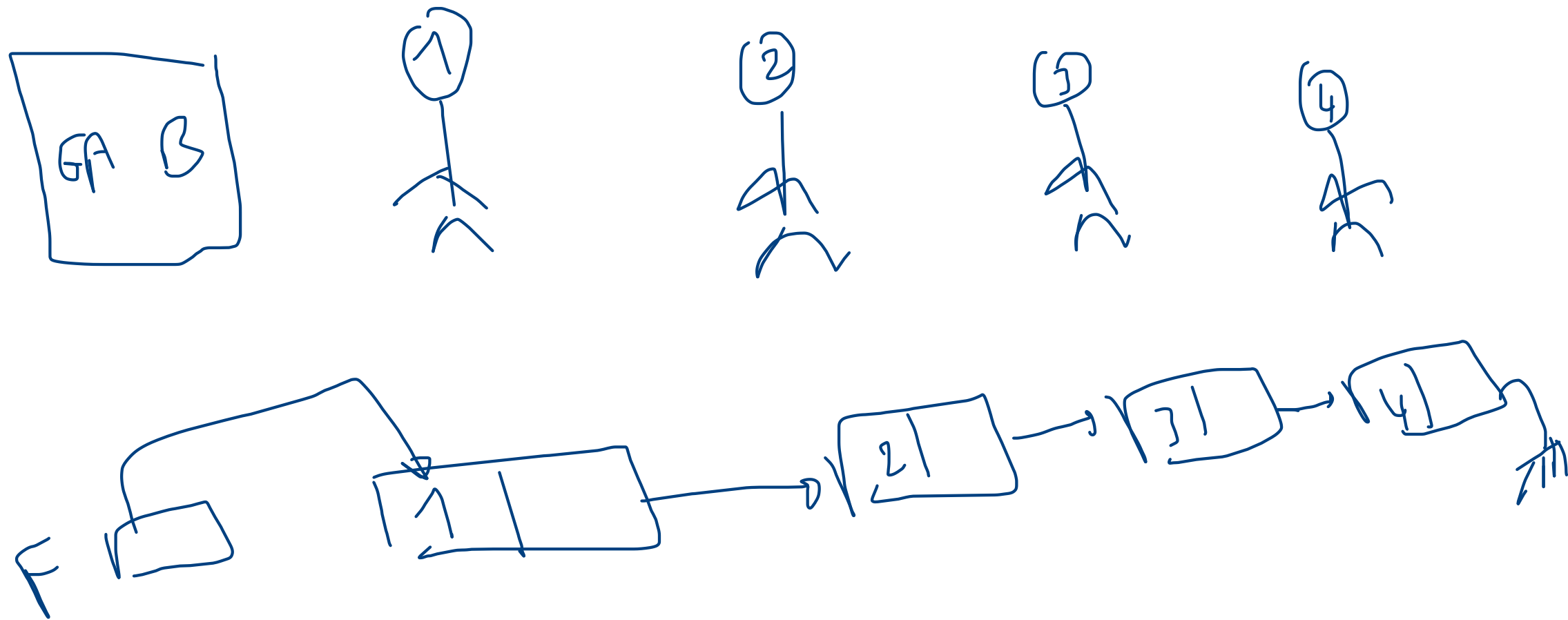
file d'attente

On ajoute à la fin

On supprime le premier



## Implémentation



Enfiler: Ajouter Fin  
Defiler: Supprimer Debut

*///Les primitives*

File FileVide()

{  
}

*////////*

int EstFileVide(File F)

{  
}

*////*

File Enfiler(File F,int x)

{  
}

*////*

File Defiler(File F)

{  
}

*////*

int PremierElement(File F)

{  
}

```
//PP
```

```
main()
```

```
{
```

```
File F;
```

```
F=FileVide();
```

```
F=Enfiler(F,1);
```

```
F=Enfiler(F,2);
```

```
F=Enfiler(F,3);
```

```
F=Enfiler(F,4);
```

```
printf("le premier element de la file est %d\n",PremierElement(F));
```

```
F=Defiler(F);
```

```
printf("le premier element de la file est %d\n",PremierElement(F));
```

```
F=Defiler(F);
```

```
printf("le premier element de la file est %d\n",PremierElement(F));
```

```
F=Defiler(F);
```

```
printf("le premier element de la file est %d\n",PremierElement(F));
```

```
F=Defiler(F);
```

```
printf("le premier element de la file est %d\n",PremierElement(F));
```

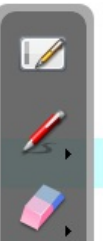
```
}
```

```
C:\Users\user\Desktop\34_2021\file\q1\q1\q1\q1.exe
le premier element de la file est 1
le premier element de la file est 2
le premier element de la file est 3
le premier element de la file est 4
la file est vide
-----
```

```

/*****On implémente une file en utilisant 2 pointeurs La structure File version statique*****/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
/// Q1 Définir le type liste
typedef struct cellule{
    int info;
    cellule * suiv;
}liste;
//Définir le type File
typedef struct t_File{
    liste *t;
    liste *q;
}File;

```



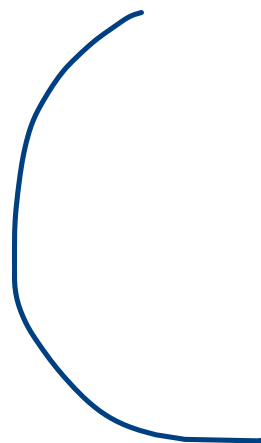
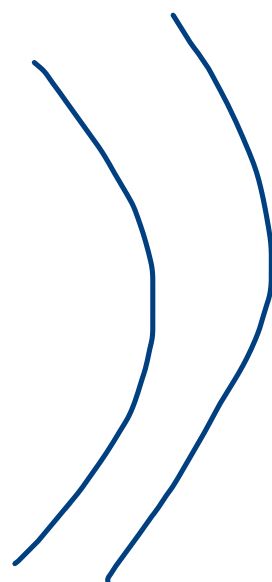
File FileVide()  
 int EstFileVide(F)  
 File Enfiler(File F,int x)  
 File Defiler(File F)  
 int PremierElt(File F)

Application n° 1 :

Verification de préférences

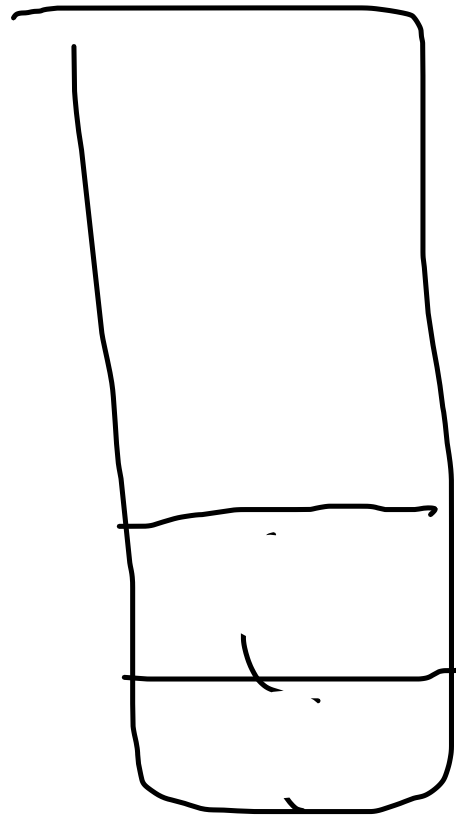
$((5+4)) \longrightarrow \text{oui}$

$(( )) \longrightarrow \text{non}$



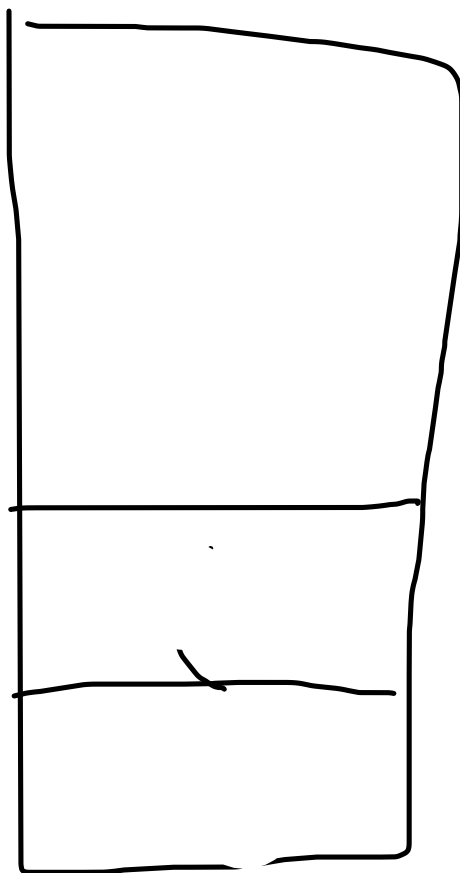
→ Fand

$$S = \left( \left( \underline{5+4} \right) * \left( \underline{5-2} \right) \right)$$



file

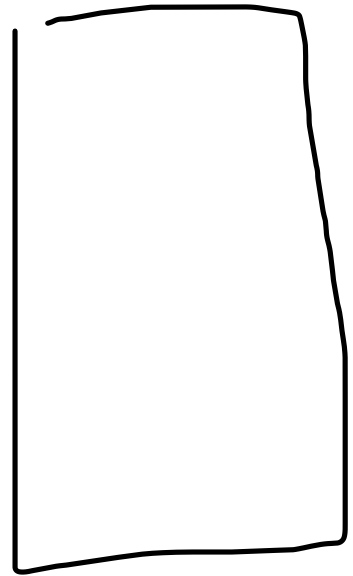
$$S = \left( \left( S + 4 \right) + 4 \right)$$



pile



S = )



f

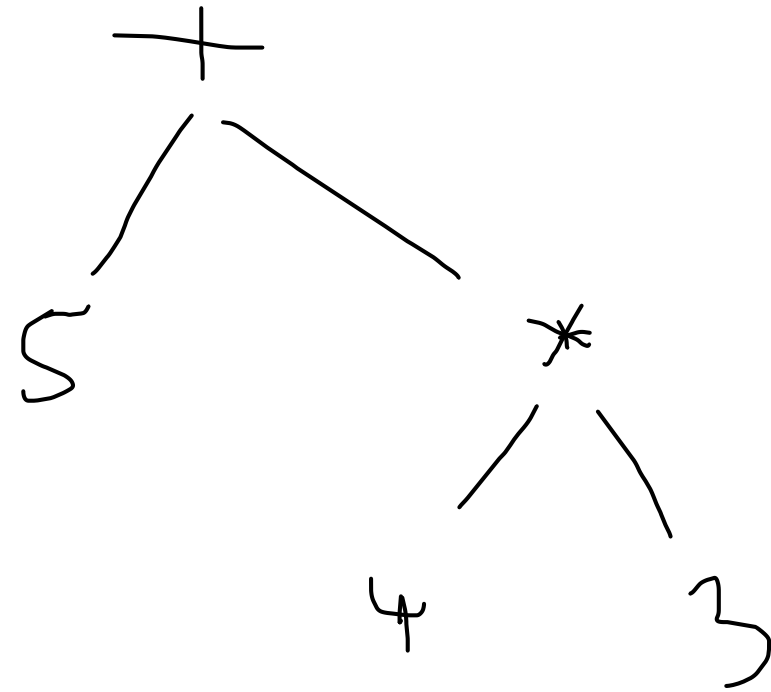
# Principe d'évaluation d'une expression arithmétique

S " 5 + 4 \* 3 "

formd.

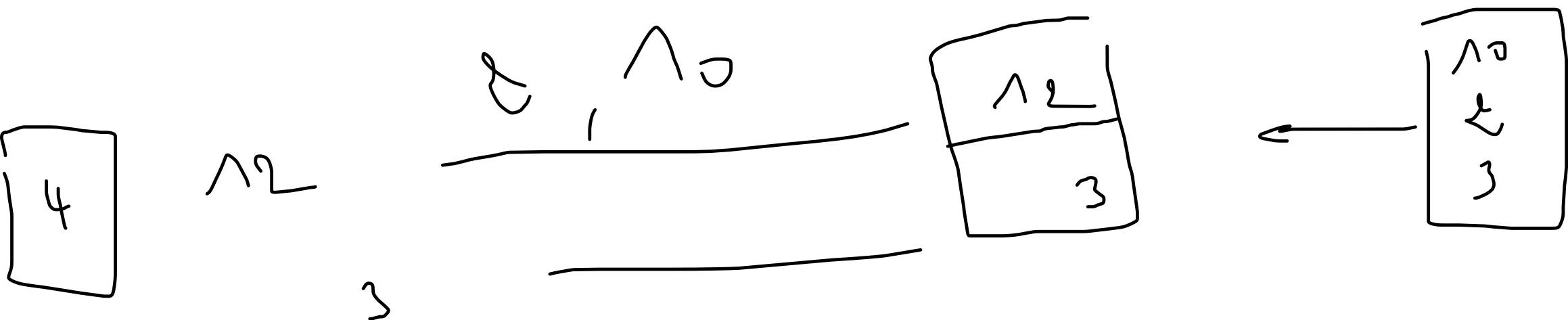
préfixe : RGD

+



$$10 + 2 \longrightarrow 0 + 102$$

$$(10 + 2) / 3 \longrightarrow / \oplus \backslash 1023$$

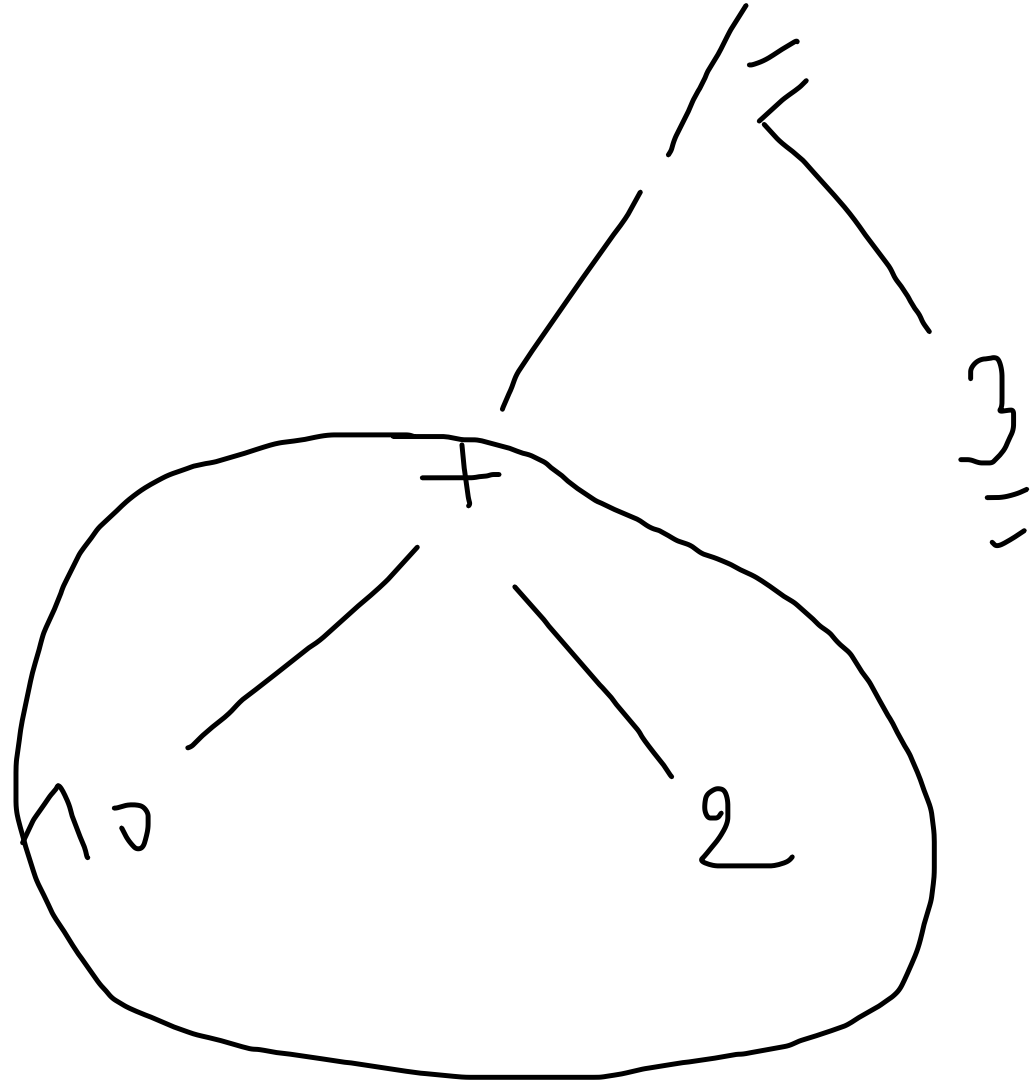


RGD

$$(10 + 2) / 3$$

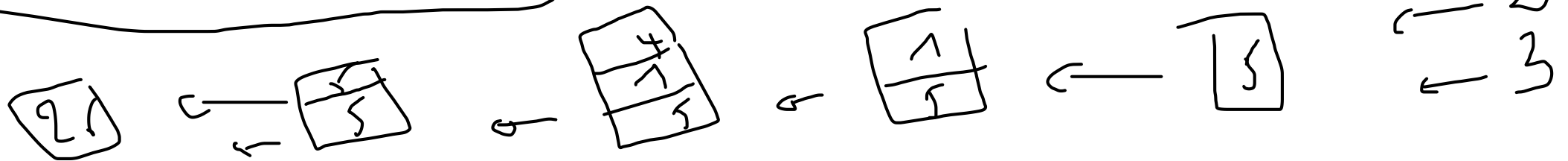
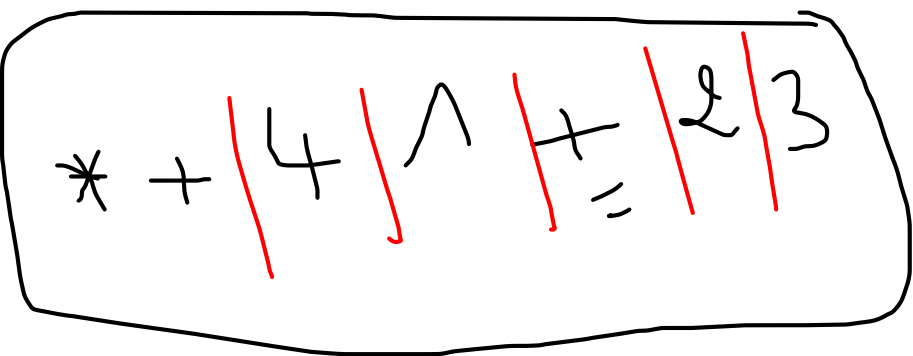
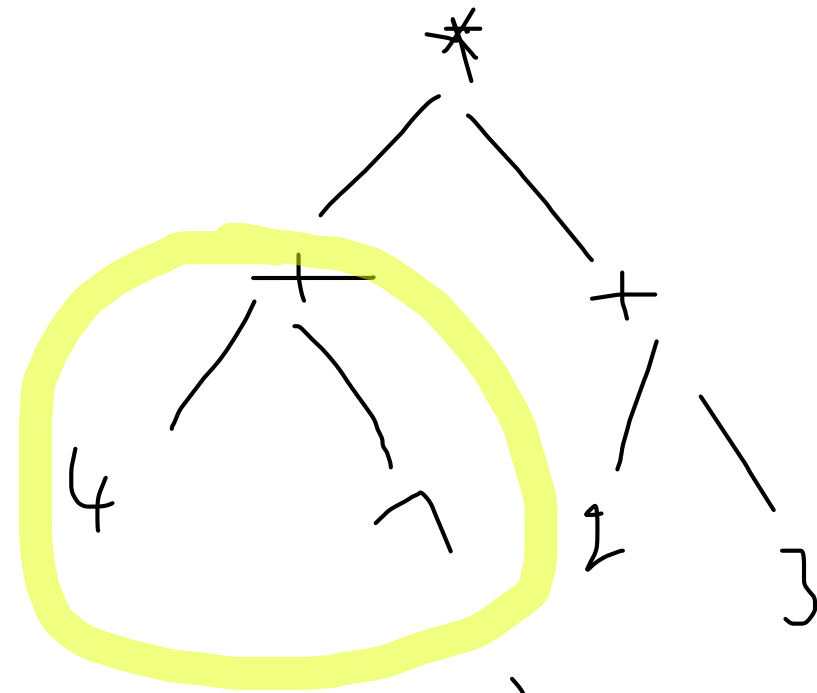
→

$$1 + 10 \leq 3$$

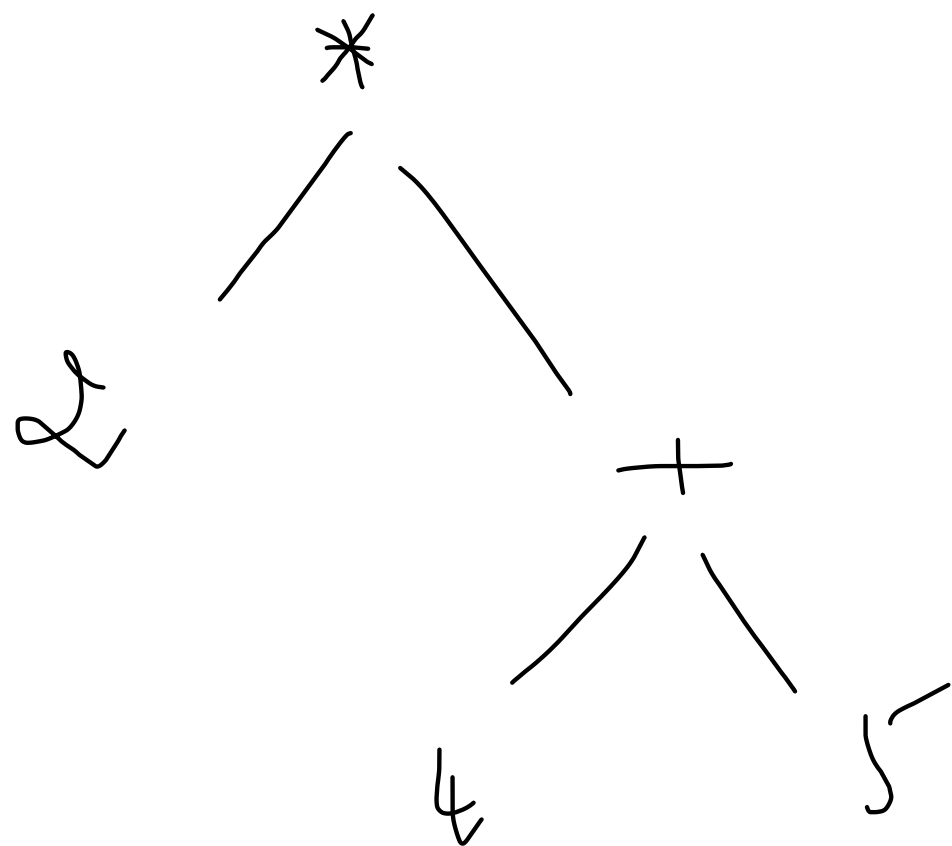


$$(4 + 1) * (2 + 3)$$

R G D



$$2 * (4 + 5) = 18$$



Proof: " \* / 2 + / 4 / 5 "



$S = "10"$

$\longrightarrow n = 10$

$S = "127"$

$\longrightarrow n = 127$

$$|b\rangle - |0\rangle = 0$$

$$|0\rangle \longrightarrow 0$$

$$|\wedge\rangle - |0\rangle$$

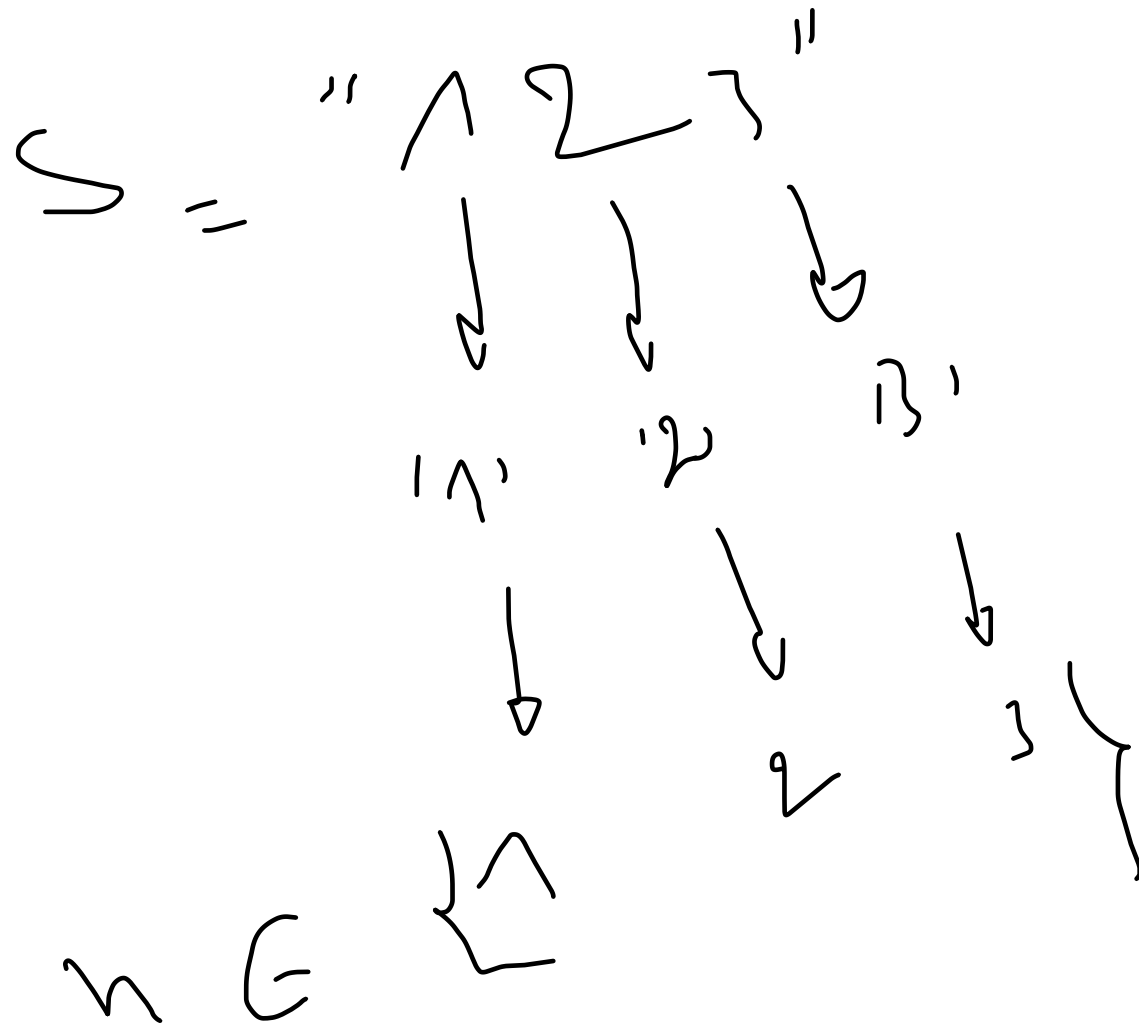
$$|\wedge\rangle \longrightarrow \wedge$$

$$|2\rangle \longrightarrow 2$$



$$n = S[i] - '0'$$

$$d = 0$$



$$S[i]$$

$$d = d * 10 + n$$

which  $\Rightarrow$

$n =$  1 2 3

$\{ r = n \% 10$   
 $=$   
 $r = r + b$

"1 2 3"

$S =$  "1 2 3"

1  
b  
1

$S[i] =$



$\text{D} \rightarrow$

$\text{D} \rightarrow$

$\text{D} \rightarrow$

$\text{D} \rightarrow$

$\text{D} \rightarrow \text{D} \rightarrow$



$\text{D} \rightarrow$