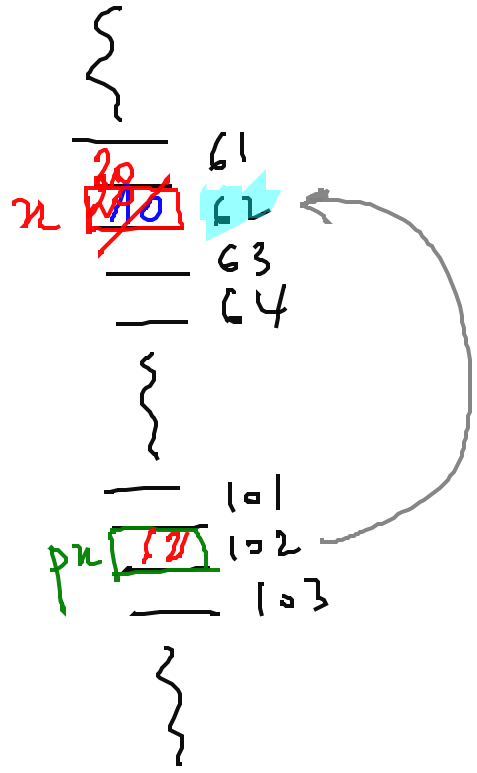
Rappel :

- Une variable est destinée à contenir une valeur du type avec lequel elle est déclarée.

```
int x;
x = 10;
```

Cette valeur de x est située à un emplacement mémoire &x ('&' adresse de)



- Un pointeur
  - c'est une variable
  - il est destiné à contenir une @
  - pour différencier une variable d'un pointeur on fait précéder la dernière par '*'

Exemple

```
int *px;
int x=10;
px = &x;
```

*px = 20 ;

px = x ;
int * int

Signaux

Ex 4.1 :

1°/ p = 7 ; légal mais dangereux

2°/ *p = 7 ; " " "

int *p ;
int y = 2 ;

3°/ p = &y ; Correcte
*p = 7 ;

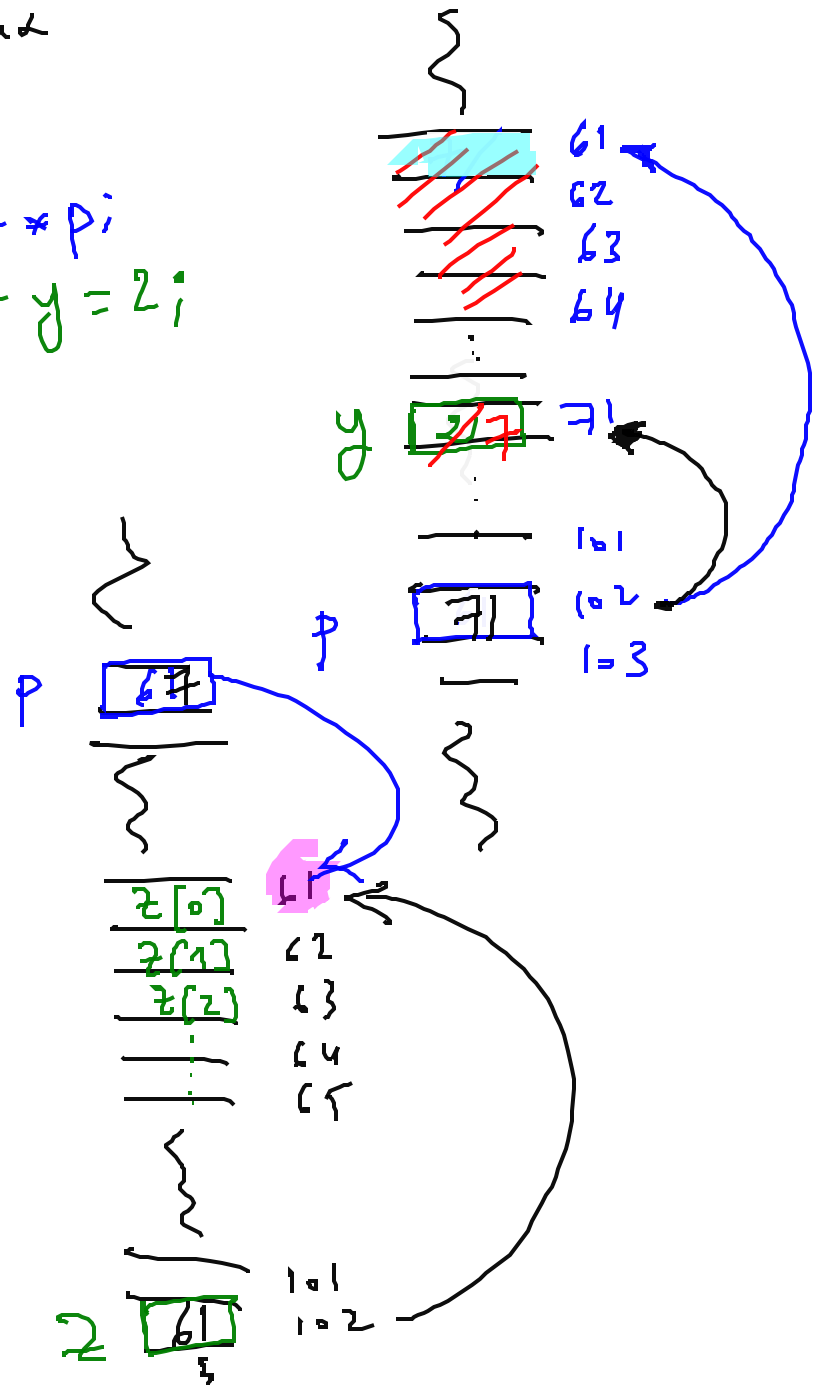
4°/ *y = *p ; Illégal

5°/ p = 2 ; Correcte
(int *) ← tableau

int z[10] ;

6°/ p = &z[6] ; Correcte

7°/ p = &z[15] ;
légal mais dangereux



8°/ p = 7 ;
p++ ;
p = 6 ; [p = p + 1 ;] légal mais dangereux,

9°/ p = &y ; p++ ; p = 6 ; légal mais dangereux ;

10°/ p++ ; p = 6 ; légal mais dangereux

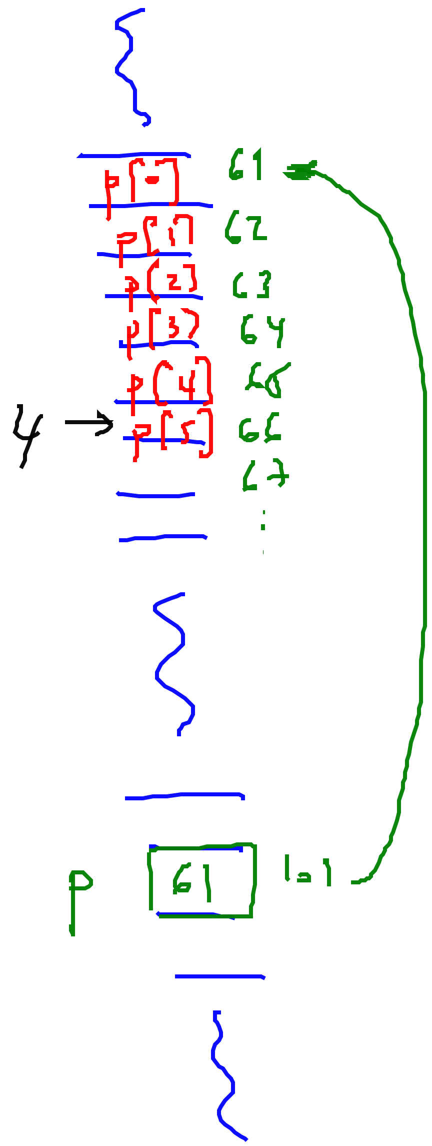
11°/ p[5] = 4 ; légal mais dangereux

12°/ p = 7 ; p[5] = 4 ; Correcte

13°/ x = *p ; Correcte

int (x)   int (*p)

int x ;



char a[] = "hello" $\overset{?}{\Longleftrightarrow}$ char *a = "hello" ;

Ex 4.2 :

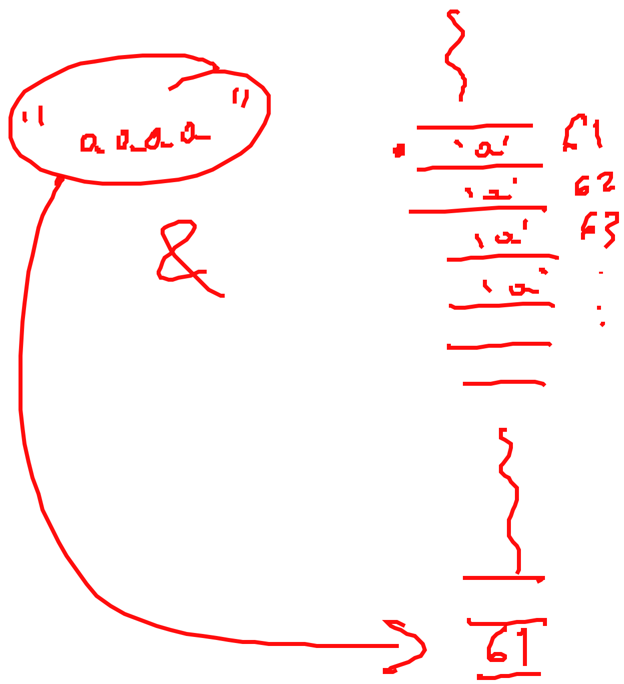
1°/ c = "a" ;

(c : char ; "a" : &)

"...." : chaîne de caractères
= tableau
= pointeur sur char

Illégale

"a"



int n ;

char c ≠ char c[20]

2°/ c = 'a' ; Correcte

' ' : 1 seul caractère

&

3°/ p = "a" ; Correcte

4°/ p = 'a' ; Illegal

Char * Char

5°/ *p = "a" ; Illegal

6°/ *p = 'a' ; Legal mais dangereux

7°/ p = "salut" ; Correcte

8°/ p = S ; Correcte

9°/ p = u ; Illégale car t est un pointeur constant

10°/ t = p ; Illégale

11°/ S = t ; Correcte

12°/ u = t ; Illégale

13°/ p = t++ ; [t++ ⟺ t = t + 1]

14°/ p = u++ ; Illegale

15°/ p = u++i ; légal mais dangereux

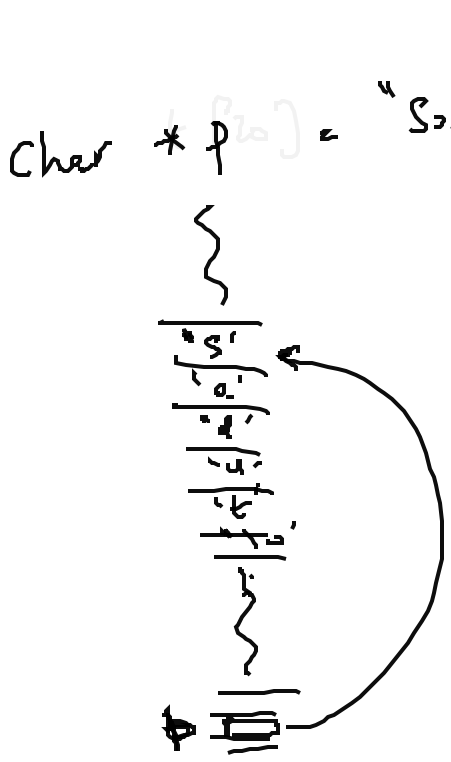
16°/ p++i ; [p = p+1] légal mais dangereux

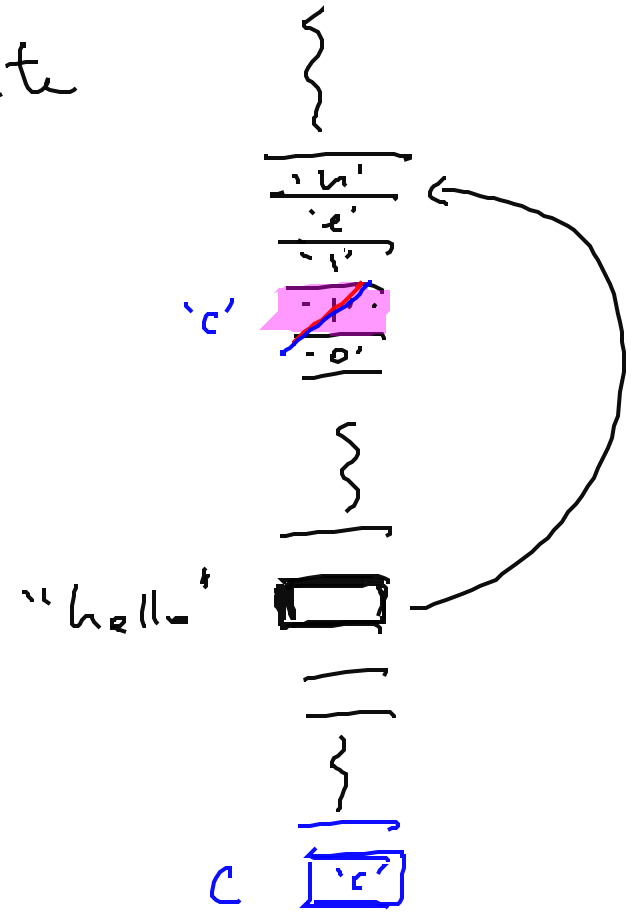
17°/ u = "hello"; u[] = "hello" illégale

18°/ t = "hello"; illégale

19°/ "hello"[3] = c; Correcte

char *p = "salut";





1°/ p = &c ; Correcte

3°/ c = *p ; Correcte

char char

l.n

(*pe).n ⟺ pe → n

Ex 4.3 :

Struct Comp

char ch[20]
double n

1°/ l.n = *p ; Correcte

double double

Struct Comp e ;

" " *pe ;

2°/ pe → n = 7.0 ; Illégal mais dangereux

double double

3°/ pe = &e ; pe → n = 7.0 ; Correcte

initialisation du pointeur

4°/ pe = malloc (struct Comp) ; Illégale

5°/ pe = (struct Comp *) malloc (struct Comp) ;

pe

n : cases

6°/ pe = (struct Comp*) malloc (sizeof (struct Comp)) ;

pointeur générique

(void *)

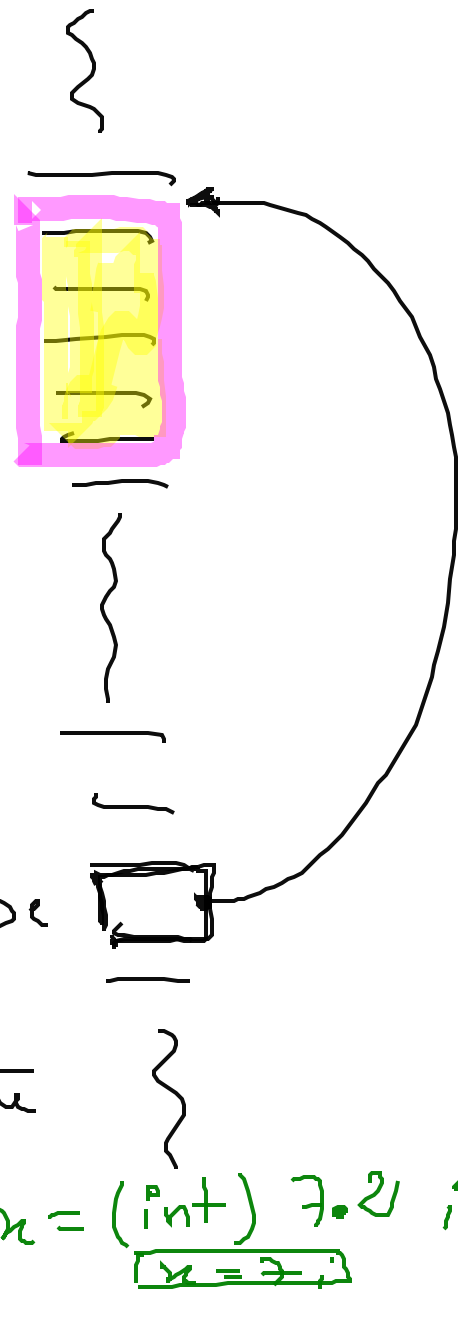
7°/ // Correcte
pe → n = 7.0 ;

8°/ // pe → t = "salut" ; Illégale

9°/ fct (pe) ; Correcte

10°/ // fct (pe) ; Correcte

11°/ // pe = (struct Comp *) 0 ; & Correcte



x = (int) 7.2 ;
x = 7 ;

12°/ " Correcte
p = &e i

13°/ " Correcte
"

14°/ p = &(pe → n) Correcte

15°/ "
p = & (pe → n) i Correcte.