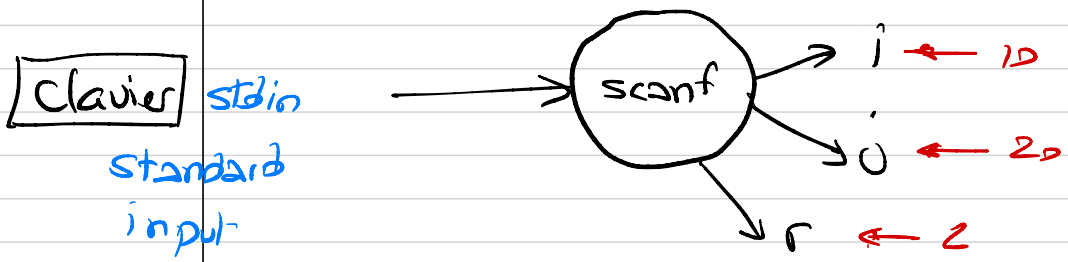
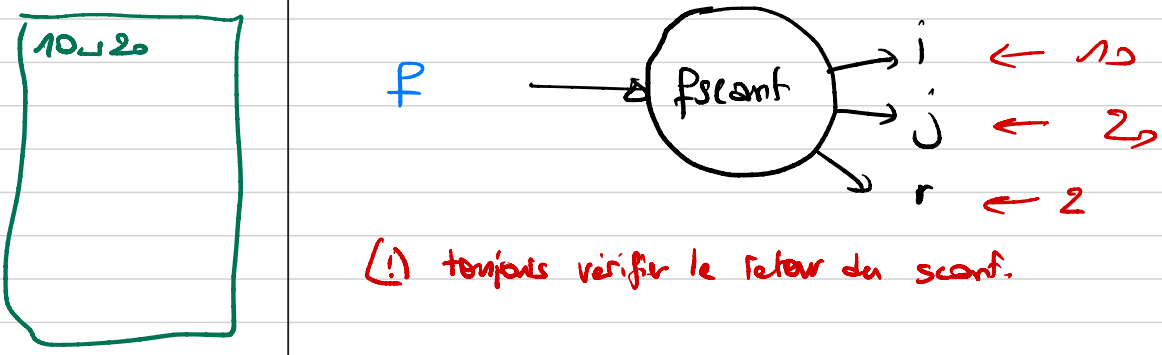
③ Accès par champs

```
r = scanf ("%d %d", &i, &j);
```



⚠ toujours vérifier le retour du scanf.

```
f = fopen ("data.txt", "r");
r = fscanf (f, "%d %d", &i, &j);
```

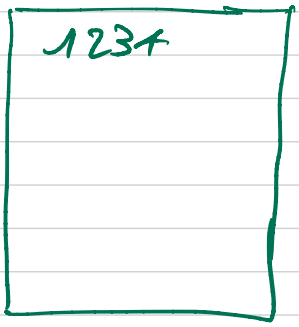


⚠ toujours vérifier le retour du scanf.

```
char * s = "10 20";
r = sscanf (s, "%d %d", &i, &j);
```

⚠ toujours vérifier le retour du scanf.

data.txt




```
int i = 42;
int j = 37;
int r = 0;

f = fopen("data.txt", "r");
```

Option 1

```
r = fscanf(f, "%d %d", &i, &j);
```

	Avant	Après
r	0	1
i	42	1234
j	37	37

Option 2

```
#define N 1024
char s[N];

fgets(s, N, f);  // s ← "1234"

r = sscanf(s, "%d %d", &i, &j);
```

r ← 1
i ← 1234
j ← 37

data.txt

```
ABCD
1234
XYZT
```

ABCD\n1234\nXYZT EOF

```
FILE *f = NULL;
f = fopen("data.txt", "r");
// test si f ≠ NULL
char s[N];

fgets(s, N, f);   s ← "ABCD\n"
fgets(s, N, f);   s ← "1234\n"
fgets(s, N, f);   s ← "XYZT"
```

	chaine	int
	"ROUGE"	1
	"bleu"	2
	"vert"	3

```
int coul = 2;

typedef enum {
    ROUGE = 1,
    BLEU = 2,
    VERT = 3
} COULEUR;

COULEUR c = ROUGE;

if ( c == VERT ) {...}
printf ("c= %d", c);        c= 1

typedef enum {
    NO_ERROR = 0,
    BAD_FORMAT = 1,
    :
} eErrorCode;
```

int

Analyse.

18266 lines (18265 sloc) | 571 KB

3 premières lignes

```
1  MeteoSchweiz/MeteoSuisse/MeteoSvizzera/MeteoSwiss
2
3  stn    time          tre200d0
4  LSN    19310101          5.3
5  LSN    19310102          5.5
6  LSN    19310103          8.4
7  LSN    19310104          5.1
8  LSN    19310105          1.4
9  LSN    19310106         -1.2
10 LSN    19310107         -2.8
11 LSN    19310108         -2.9
12 LSN    19310109         -5.0
13 LSN    19310110         -5.9
14 LSN    19310111         -6.8
15 LSN    19310112         -5.1
16 LSN    19310113         -3.3
```

(1) "LSN 19310101 5.3\n"

année

temp

(2) Comment faire

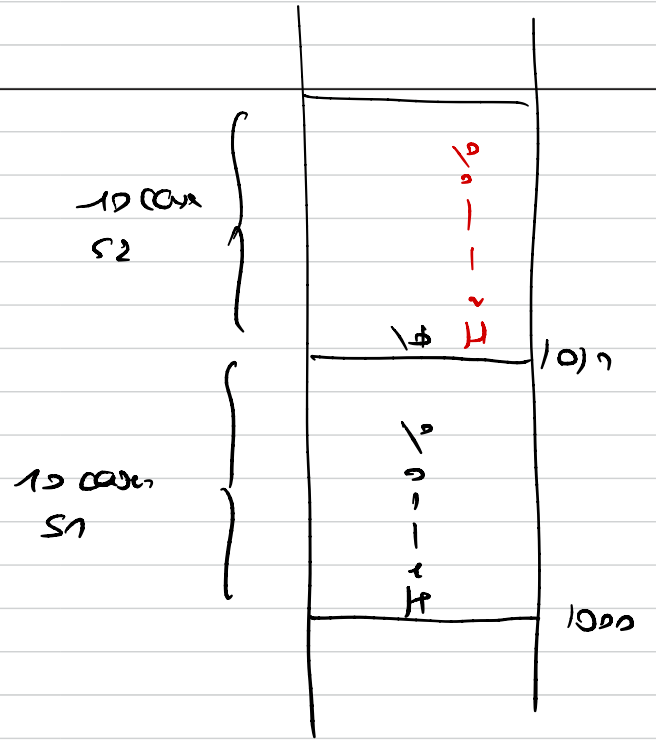
1) l'accumulation des temp pour un même année

2) les transitions pour les années

(3) Calcul valeur moyenne et affichage.

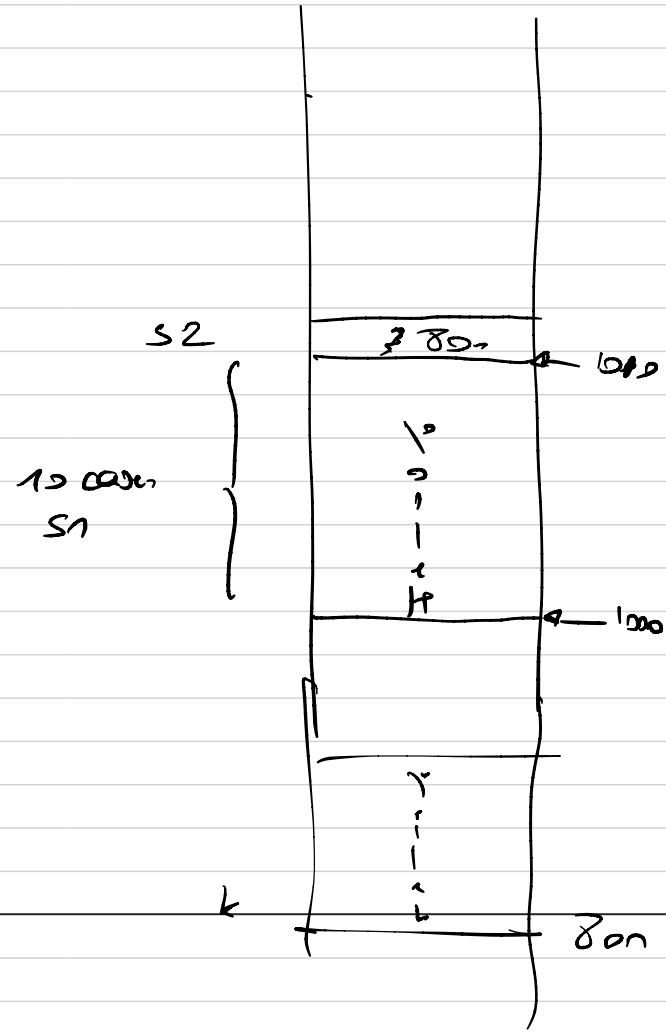
```
char  s1[10] = "Hello";
char  s2[10] = "";

strcpy(s2, s1);
```




```
loop  k = 28
char  s1[10] = "Hello";
char  *s2;  = NULL;

strcpy(s2, s1);
```



int f(void) {

==

return [ ] int ;

}

void f(void) {

==

return;

}

./app ~

usage:

./app

./app

main:

usage(argv[0]);

"./app"

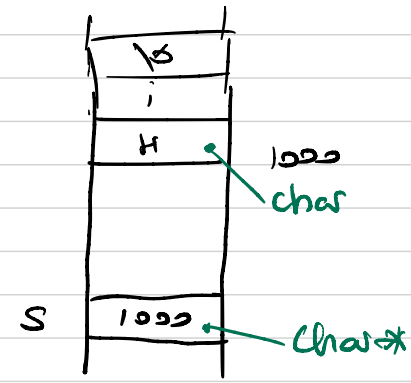
usage(char *appName) {

printf("%s ...", appName);

chaines et pointeurs, quel bonheur !

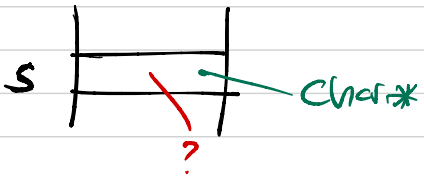
11. III. 2021

① `char *s = "Hi";`

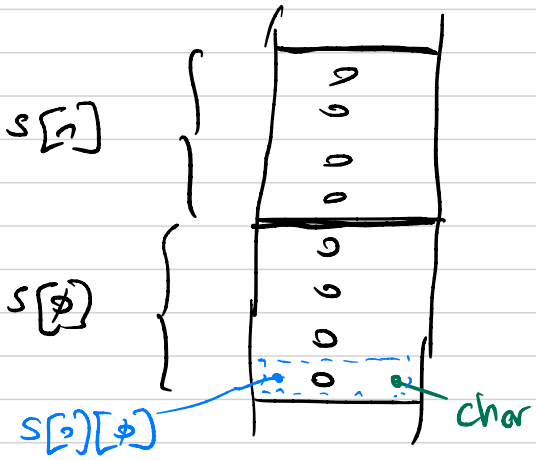


② `char *s;`

(!) valeur indéfinie. danger ...



③ `char s[2][4] = {0};`



④ `char *s[2] = { "ici", "la" };`