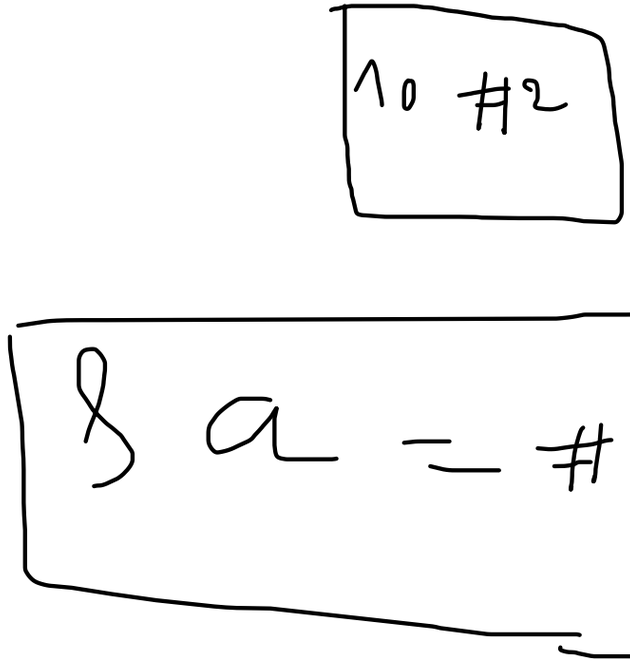
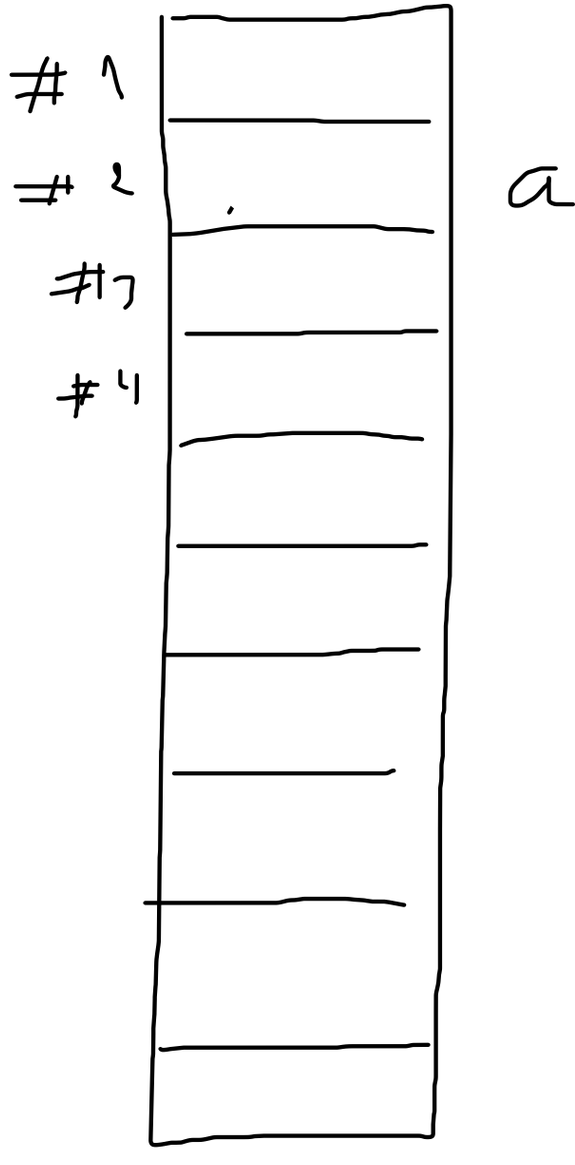
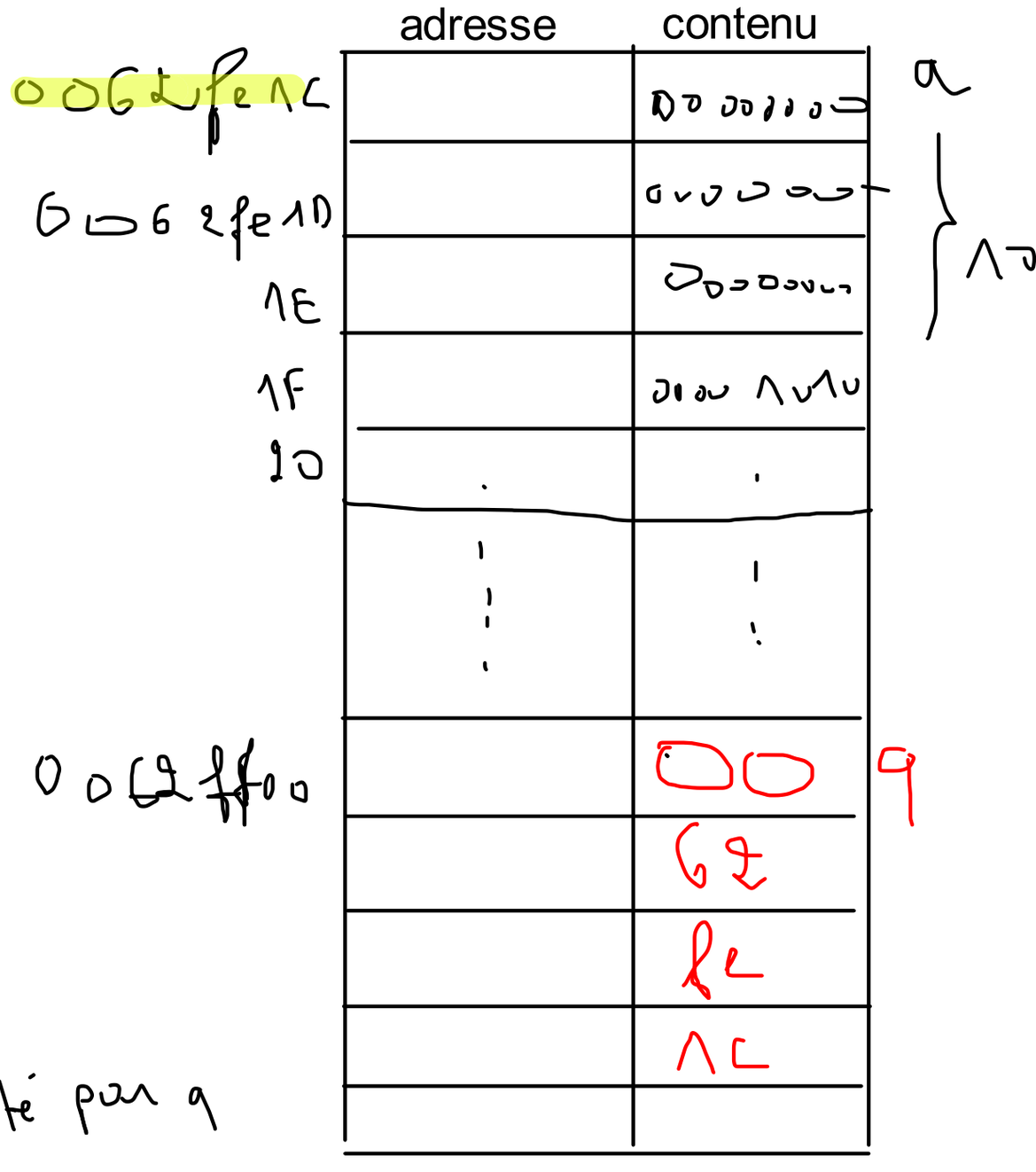
Chapitre 2. Les pointeurs

```
main()
{
int a;
a=10;
printf("a=%d",a)
printf("l'adresse de a %x",&a);
}
```





```
main()
{
int a;
int *q; //pointeur
a=10;
printf("a=%d",a);
p=&a;
printf("l'adresse de a %x",&a);
printf("la, valeur de p %p",q);
}
```



&a = 0062fe1C

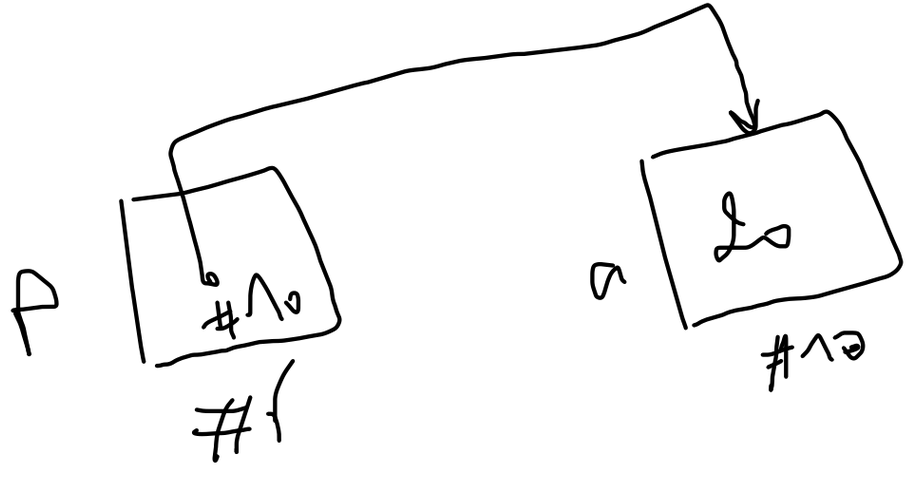
q = &a

*q = le contenu pointé par q

Définition:
un poin teur est une variable qui contient l'adresse d'une autre variable

int a, *p;

p = &a



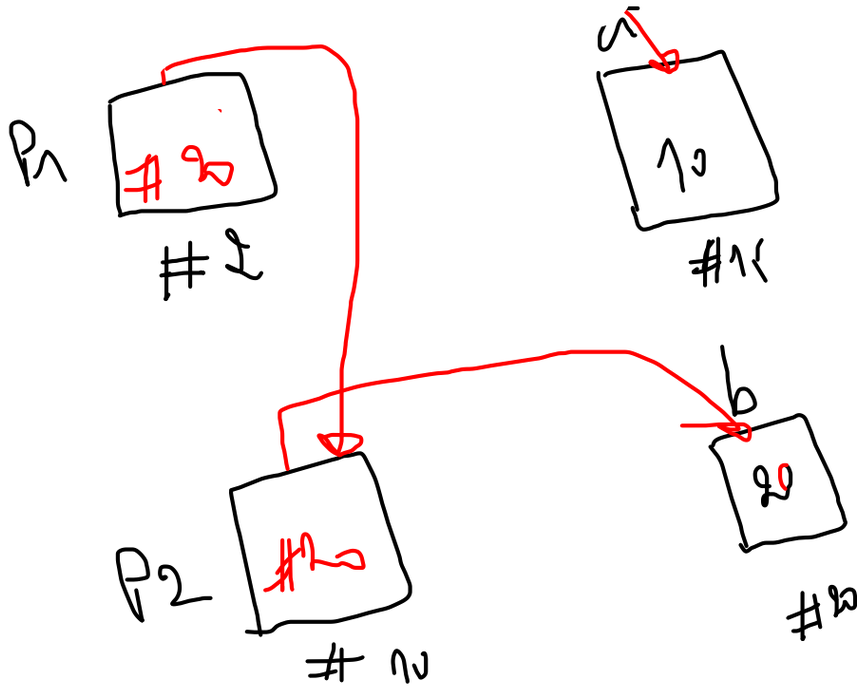
*p ⇔ a

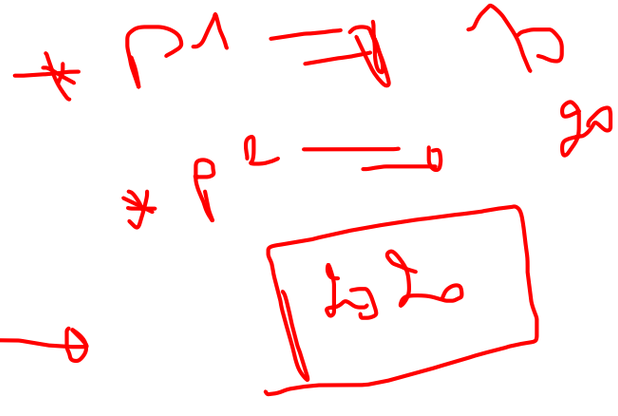
&p = #f

&a = #13

p = #10

```c
#include<stdio.h>
main()
{
	float a,b,*p1,*p2;
	a=10;
	b=20;
	p1=&a;
	p2=&b;
	printf("la valeur de p1 est %p\n",p1);
	printf("le contenu pointé par p1 est %f\n",*p1);
	printf("la valeur de p2 est %p\n",p2);
	printf("le contenu pointé par p2 est %f\n",*p2);
	p1=p2;
	print("%f %f",*p1,*p2);
}
```





Les opérations arithmétiques sur les pointeurs

```
int a, *p;
```

$a = 10$

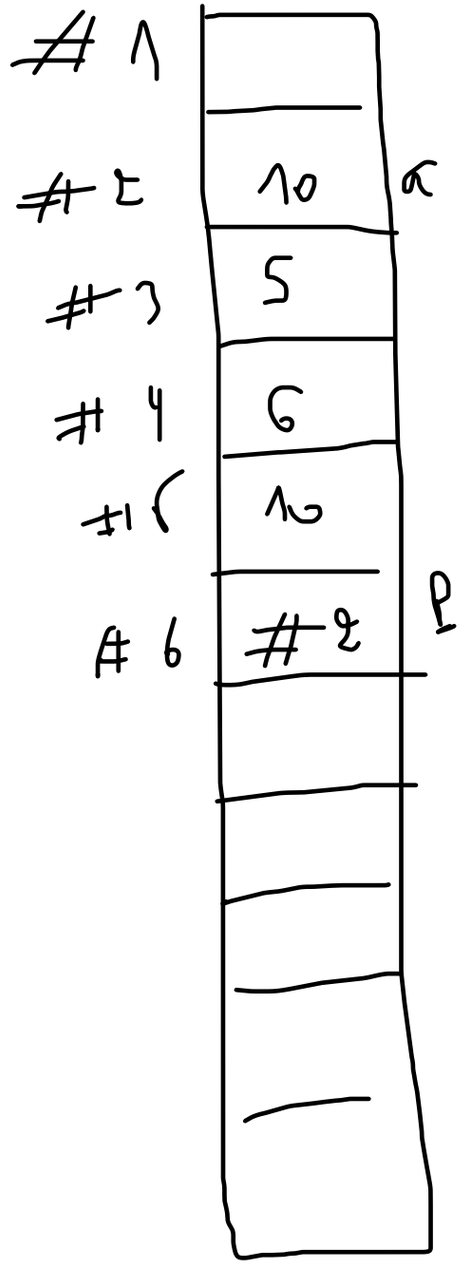
$p = \&a$

$p + 1 = \#3$

$*(p+1) = 5$

$p + 2 = \#4$

$*p + 2 = 10 + 2 = 12$



La régle:

type *p

p + i = p + i x sizeof(type)

sizeof(char)=1
sizeof(float)=4
sizeof(int)=4

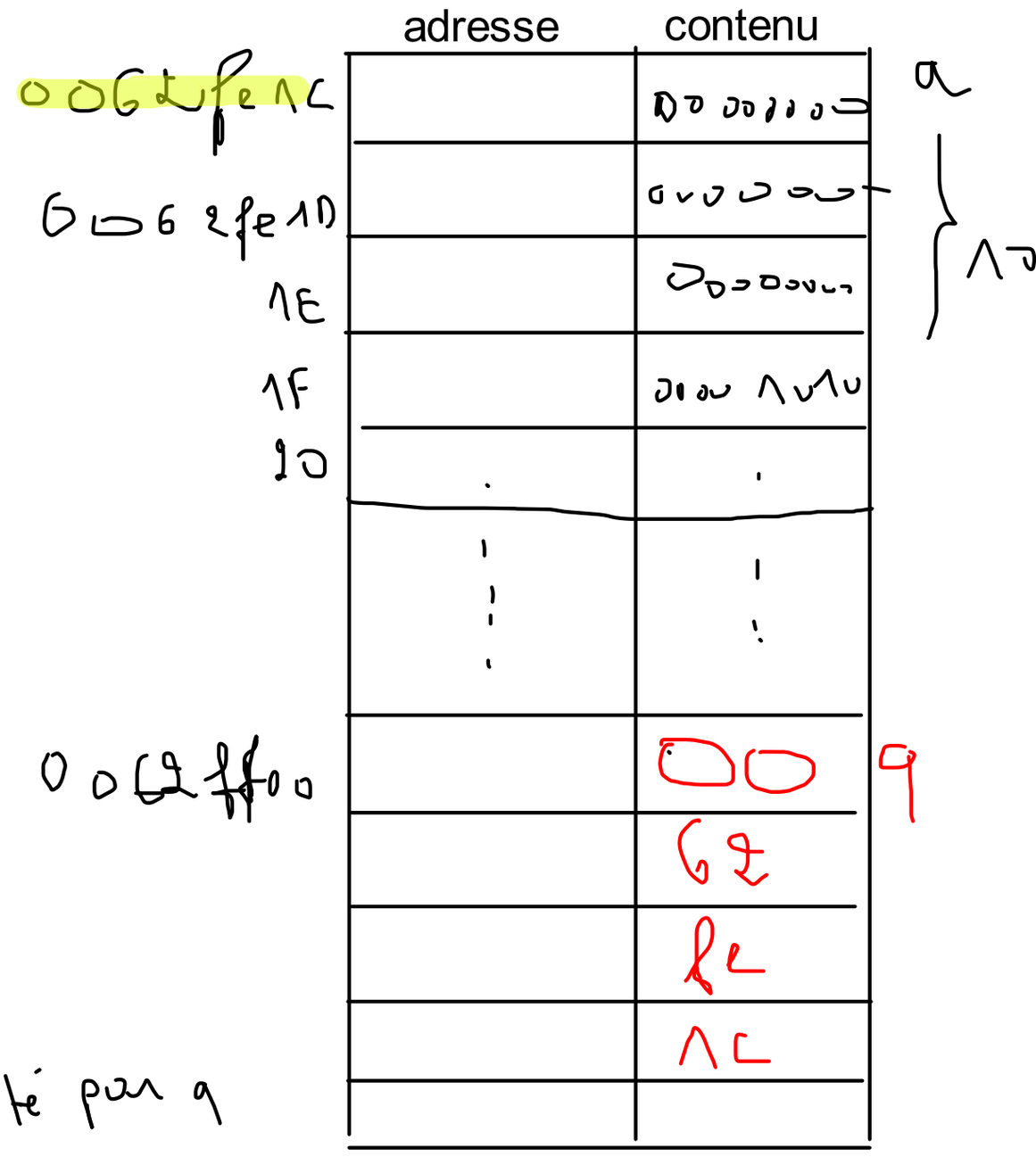
```
int a,*q;
a=10;
q=&a;
```

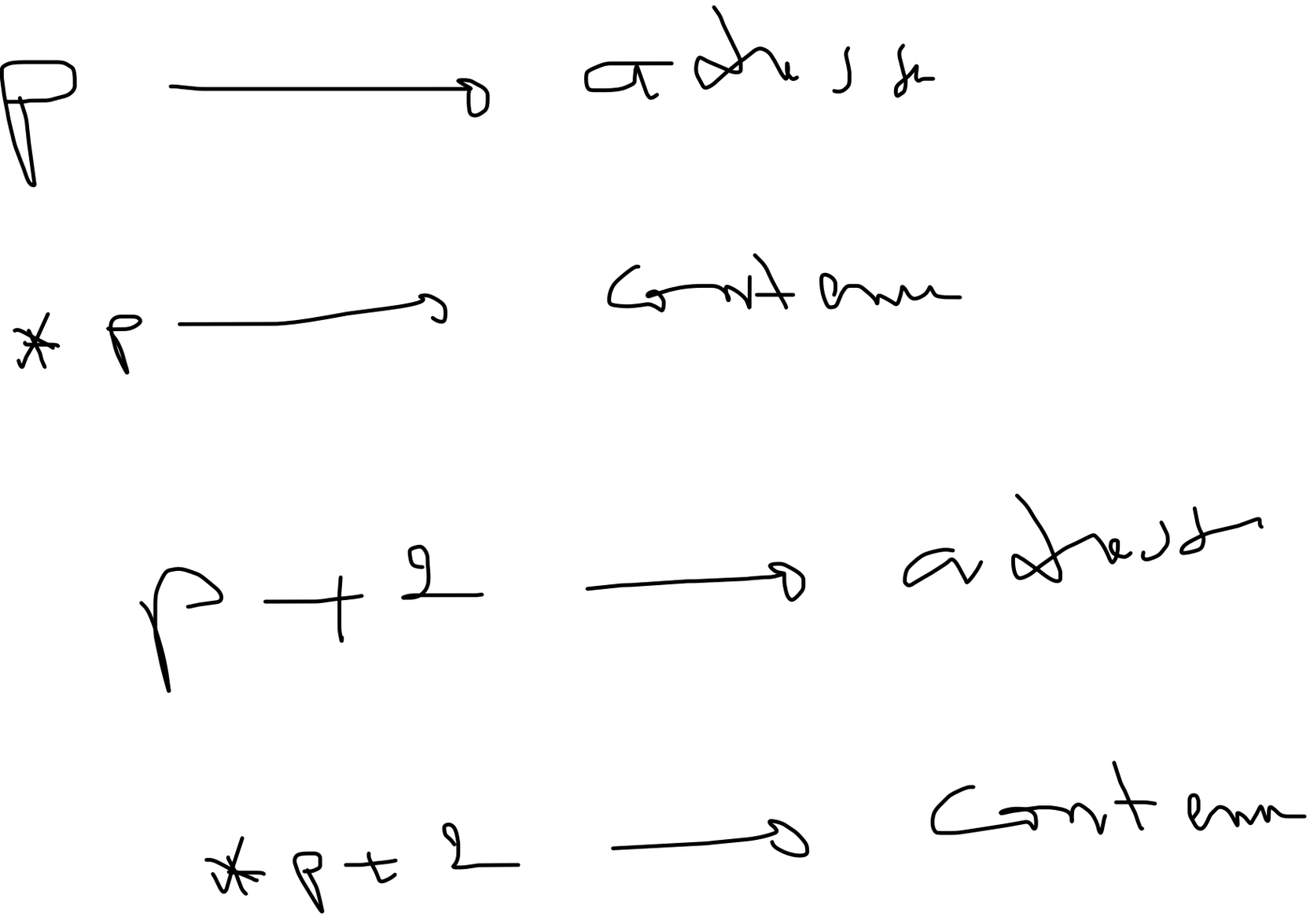
$q = 0062fe1c$

$q+1 = 0062fe1c+1\times4$

$= 0062fe20$

$q+2 = 0062fe24$



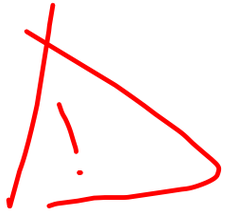


int a, *p1, *p2, *p3

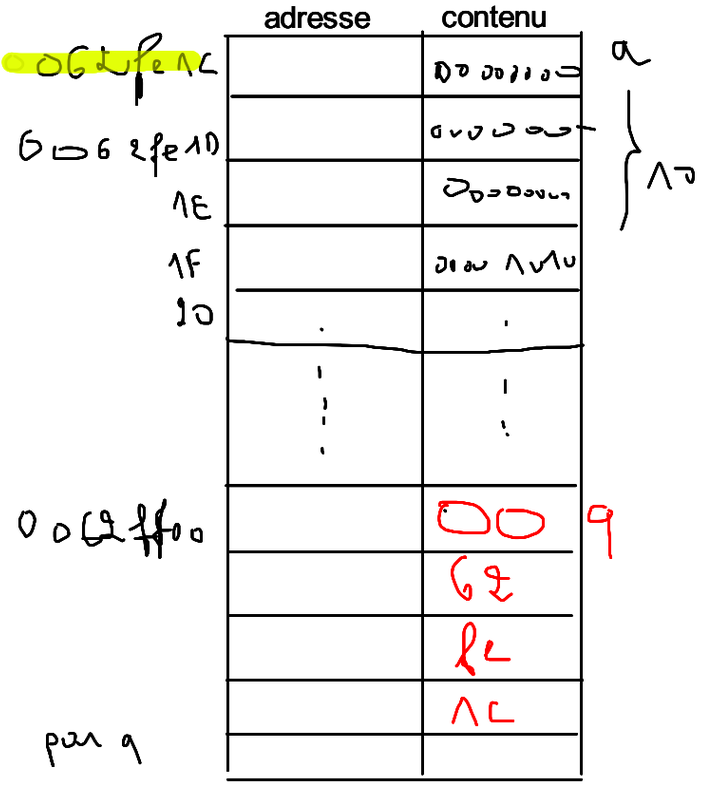
p1 = &a

p2 = p1 + 2

p3 = p1 + p2

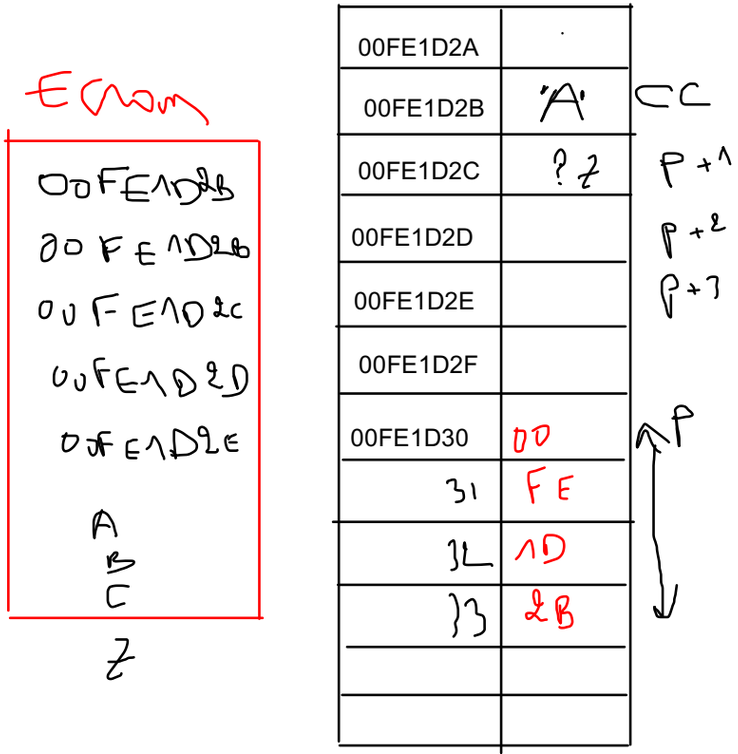


improvigh




```
f#include<stdio.h>
main()
{
	char cc,*p;
	cc='A';
	p=&cc;
	printf("l'adresse de cc est %x\n",&cc);
	printf("la valeur de p est %p\n",p);
	printf("la valeur de p+1 est %p\n",p+1);
	printf("la valeur de p+2 est %p\n",p+2);
	printf("la valeur de p+3 est %p\n",p+3);
	printf("la valeur est %c\n",*p);
	printf("la valeur est %c\n",*p+1);
	printf("la valeur est %c\n",*p+2);
	printf("la valeur est %c\n",*(p+1));

}
```



Les pointeurs et les tableaux

int T[5] = {2, 4, 6, 8, 10}

int *pi

p = T

*p = 2

p+1 = &T[1]

*(p+1) = T[1] = 4

*p+1 = 2+1 = 3

p+2 = &T[2]

⚠ T = &T[0]

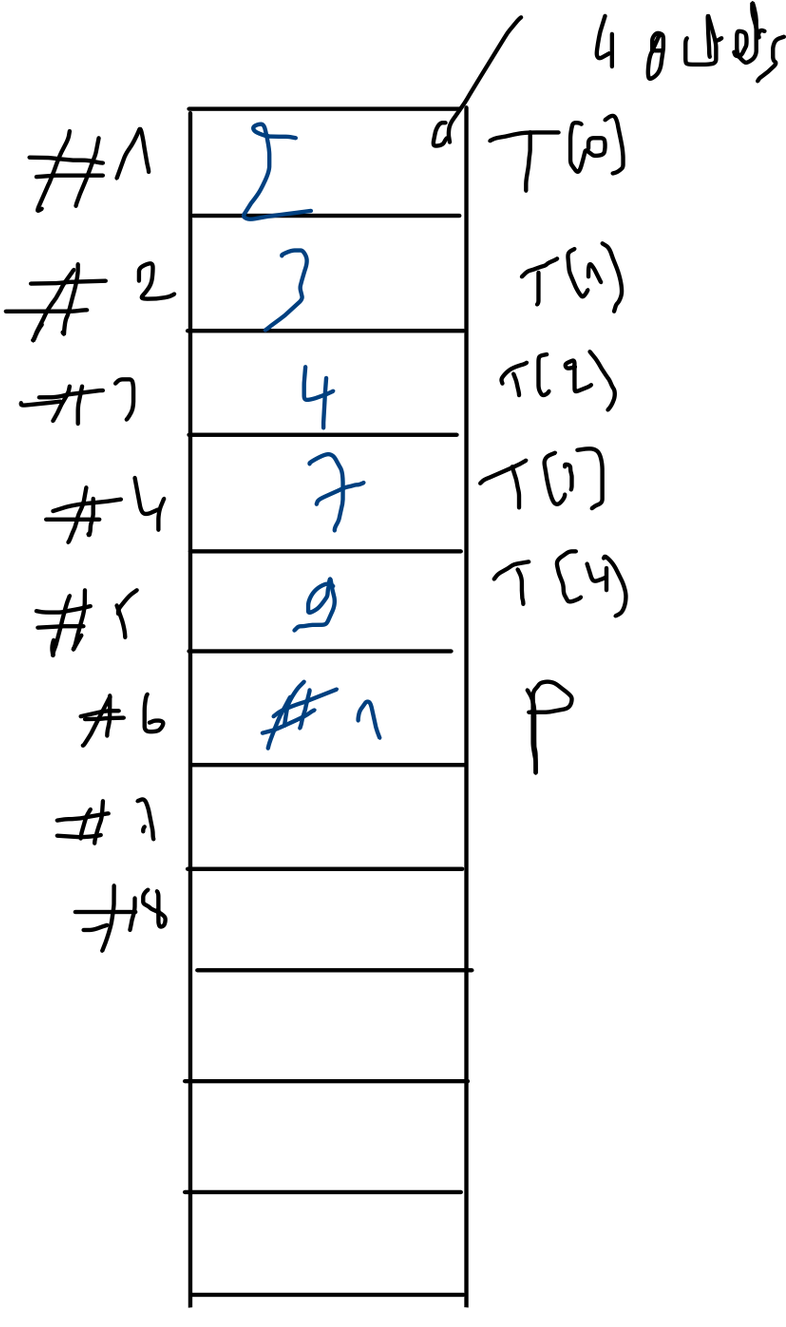
p+i = p + i * sizeof(type)

⇓

T+i = T + i * sizeof(type)

*p = *T = T[0]

int T[5]={2,3,4,7,9},*p;
p=&T[0]



p+1 = &T[0] + 1x4

= #1 + 1 = #2 = &T[1]

p+2 = #1 + 2 = #3 = &T[2]

*(p+1) = 3 = T[1]

Si T est un tableau

et on fait p = T

p + i = T + i = &T[i]

*(p+i) = *(T+i) = T[i]

Remarque:

un tableau est un pointeur

```c
#include<stdio.h>
main()
{
	int T[5],i;
	//remplir le tableau
	for(i=0;i<5;i++)
	{
	printf("T[%d]=",i);
	scanf("%d",T+i);
	}
	//Affichage
	for(i=0;i<5;i++)
	printf("%d\n",*(T+i));

}
```