A little C reminder

Noël De Palma
UJF

Basic types

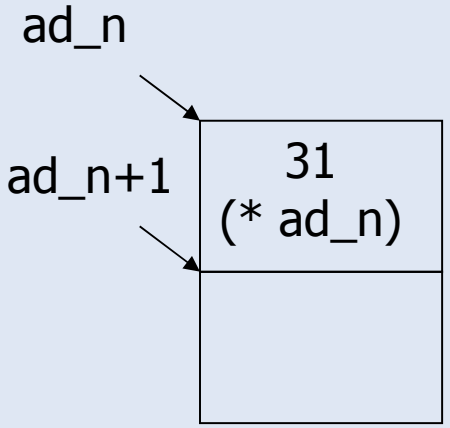
- Integer types
  - short/int/long/double/long double
    - Signed by default
  - Unsigned
    - No sign bit
- Character Type
  - char

Array

- Array declaration (1st form)
  - *Type arrayName [size]*
  - Ex :
    - int t0[20]
      - one dimension
      - Index range : 0 à 19. (80 octets pre allocated memory)
      - t0[4] : access to the fift element
    - int t1[10][20]
      - 2 dimensions
      - Index range : 0 à 9 and 0 à 19 (800 octets octets pre allocated memory)
      - T1[2][3]
    - Init :
      - int t0 [3] = {2,1,4}

Pointers

- A pointer = an address and a type
  - Declaration
    - Type * ptr;
  - int * p0; // ptr to an int
  - Access to the pointed value
    - *ptr
  - Example

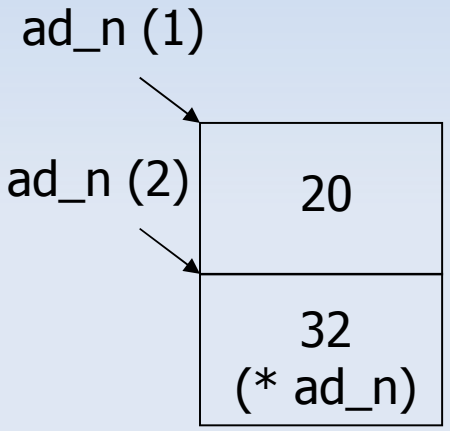



```
int n = 20;
int * ad_n;
ad_n = & n; // get the @ of n
*ad_n = 30; // n=30
(*ad_n)++; // n=31
*ad_n++; // !!! Return the value at the address &n + 4
```

Pointers

```
int n = 20
int * ad_n = &n; // (1)
*(ad_n +1) = 32; // (2)

// ad_n is a pointer to an integer.
// ad_n+1 = @ of the next integer !!!
// si ad_n  = ox00000000 then
// ad_n+1 = ox00000004 (an int is coded on 4
   octets !!!)
```



Array again

- An array's name is a pointer
  - int t[20]
  - The notation ***t*** is equal to ***&t[0]***
  - t + 1 // &t[1]
  - t + i // &t[i]
  - t[i] // *(t+i)
  - *(t+2) = 3; // t[2] = 3

Array again

Int t0[20];

can be written :

*int *t0;*
*t0 = malloc(20 * sizeof(int));*

Access :

**t0 = 3; // t0[0] = 3*
**(t0+1) = 4; // t0[1] = 4*
*t0[2] = 5; // *(t0+2) = 5*

Function parameter

- Pass by value

```
void swap(int a, int b){
   int sv;
   sv = a; a = b; b = sv;
    printf(”swap a : %d, b : %d”, a,b);
}

main() {
   int a=5; int b = 10;

   printf(”a : %d, b : %d”, a,b); // a : 5, b : 10
    swap (a,b);                   // a : 10, b :5
   printf(”a : %d, b : %d”, a,b); // a : 5, b : 10
}
```

Function parameter

- Pass by address

```
void swap(int *a, int *b){
   int sv;
   sv = *a; *a = *b; *b = sv;
    printf(”swap a : %d, b : %d”, *a,*b);
}

main() {
   int a=5; int b = 10;

   printf(”a : %d, b : %d”, a,b); // a : 5, b : 10
    swap (&a,&b);                  // a : 10, b :5
   printf(”a : %d, b : %d”, a,b); // a : 10, b : 5
}
```

Structure

- A point

```
struct point {
   int x;
   int y; };
struct point p0;
p0.x = 2; p0.y = 3;
```

- A list of points

```
struct list_point {
   int x; int y;
    struct list_point *next } ;
struct list_point *p; p->x = 2; p = p->next;
```