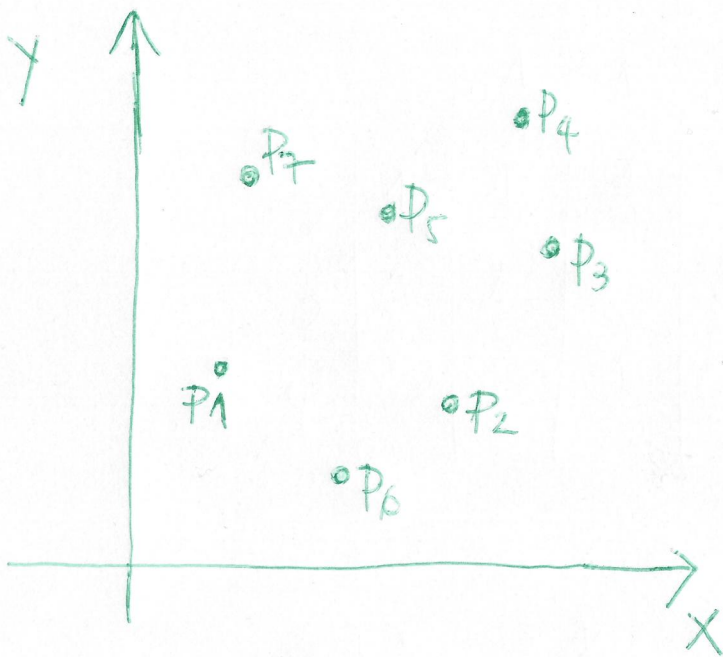


REGISTROS (STRUCT)



Cómo representarlos en un programa?

(1)

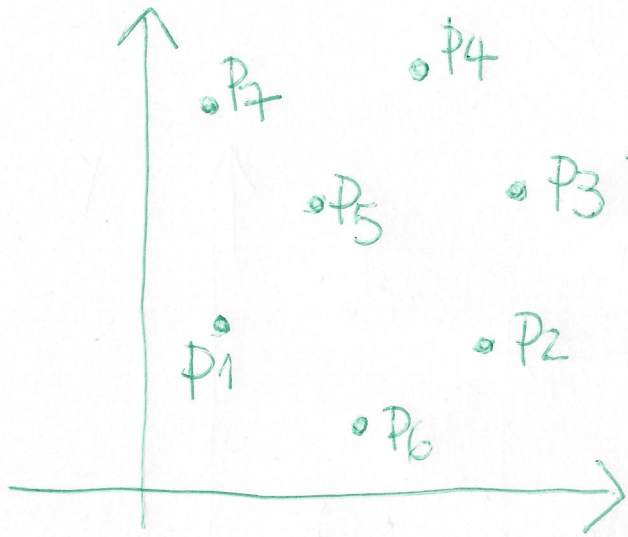
```
int x1, x2, x3, x4, x5, x6, x7;  
int y1, y2, y3, y4, y5, y6, y7;
```

Muy MALA Decisión! ▽
0

(2)

```
int x[7];  
int y[7];
```

un poco mejor... pero



Un punto es un objeto
que puede ser descrito
usando 2 atributos

↳ coord x
↳ coord y

```
typedef struct {
```

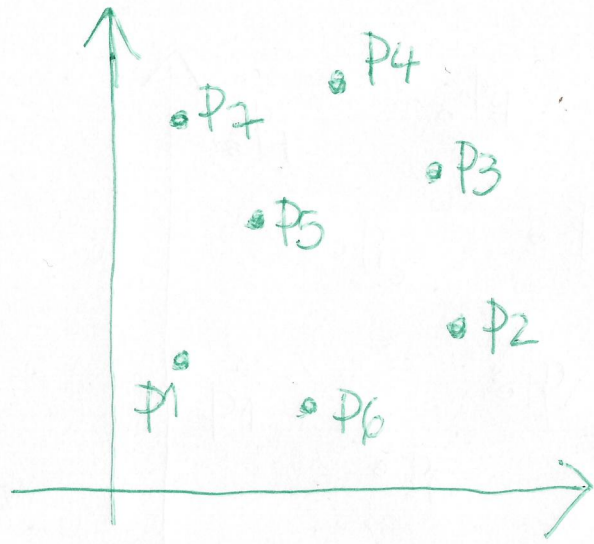
```
    int x;  
    int y;
```

```
} punto;
```

punto p1, p2;

ES un tipo de datos
definido por quien codifica

VER código fuente
en pag de la
asignatura



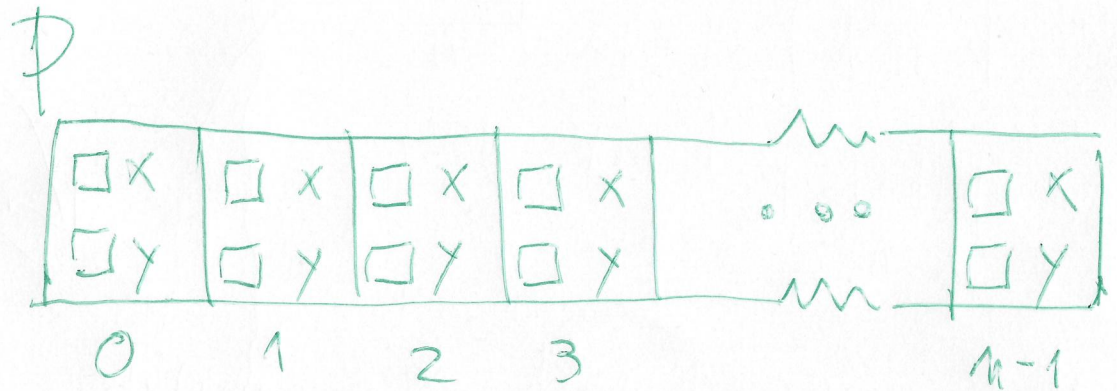
Dado que la cantidad de puntos puede ser variable, es mejor definir un arreglo de puntos

(1) definir el tipo de datos

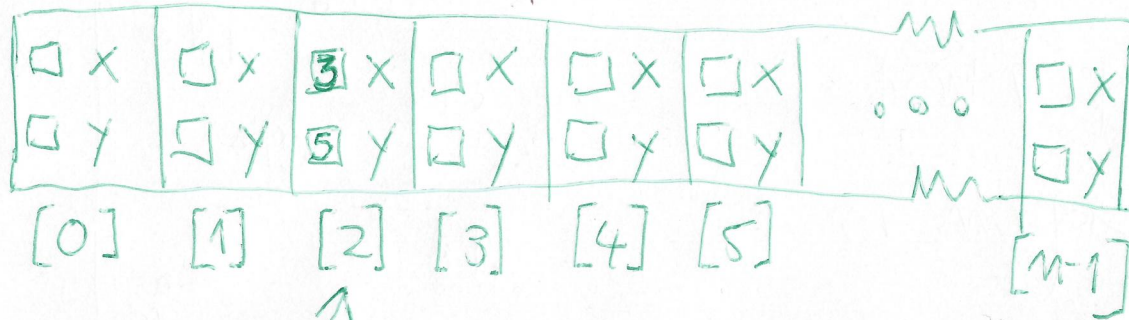
```
typedef struct {
    int x;
    int y;
} punto;
```

(2) definir el arreglo:

punto P[n];



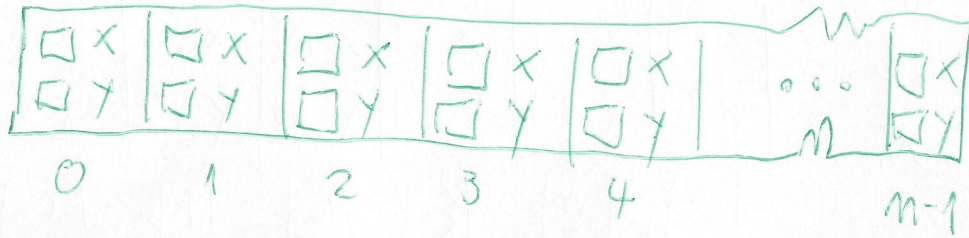
Ⓟ



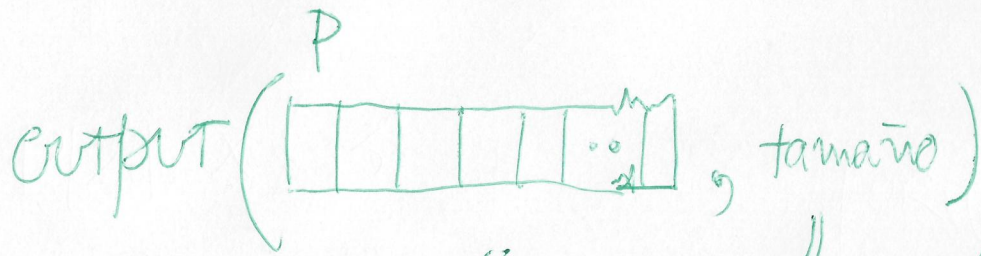
P[2].x = 3
P[2].y = 5

```
for (i=0 ; i < n ; i++)  
scanf("%d %d", &P[i].x, &P[i].y)
```

P



Punto $p[n]$;



output (punto *, int)

El código de la función output.

```
output ( punto *p, int n ) {  
    int i;  
    for ( i = 0; i < n; i++ ) {  
        printf ( "%d\n", p[i].x );  
        printf ( "%d\n", p[i].y );  
    }  
}
```