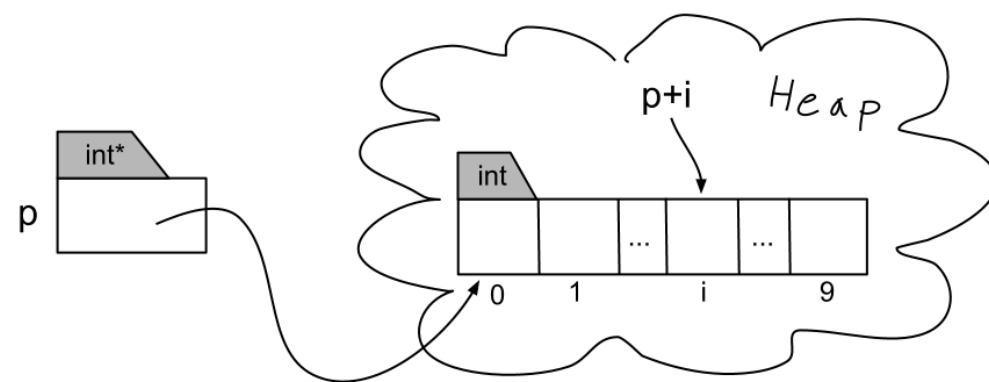




Profesor JTP
MCP. Miguel Silva

PUNTEROS



Profesores Titulares
Ing. Israel Pavelek
Ing. Behringer Alejandro



¿Qué es un Puntero?

¿Y cómo se declara?

Es una Variable
que contiene
direcciones de
memoria

<< Tipo de dato >> * identificador de puntero

Ejemplo

int a // defino una variable

int *p // define un puntero a un entero p

p=&i // asigna la dirección de i a
p



¿Operadores de Puntero?



& Obtiene la dirección de memoria de una variable

* En la definición de la variable, indique que la misma será un puntero del tipo que dato que lo definan

* En el cuerpo de un programa, se utiliza para para obtener el contenido de una variable de tipo puntero



¿Ejemplos de Puntero?

```
int i, a;  
int *ptr;  
printf("Ingrese un valor de i:");  
scanf("%d", &i);  
printf("Ingrese un valor de a:");  
scanf("%d", &a);  
ptr = &i;
```



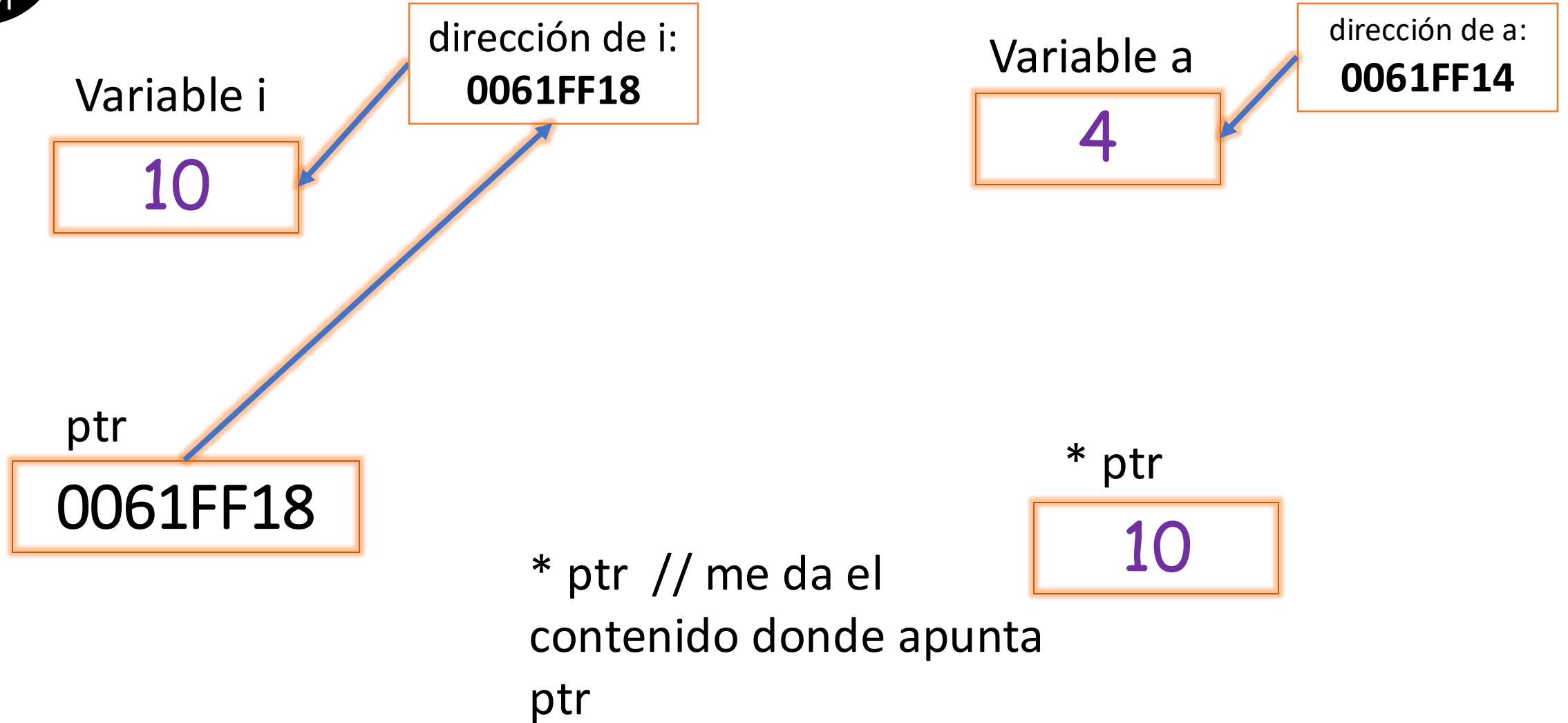
¿Ejemplos de Puntero?

```
printf("contenido de i=%d\n", i);  
printf("contenido de a=%d\n", a);  
printf("contenido de ptr=%d y direccion de ptr%p\n", *ptr, (void  
*)ptr);  
printf("direccion de a: %p\n", (void *)&a);  
printf("direccion de i: %p\n", (void *)&i);  
printf("direccion de ptr: %p\n", (void *)ptr);  
return 0;
```

En C, void* es un tipo de puntero genérico. Puede apuntar a cualquier tipo de dato, ya sea un int, float, char, o cualquier otro tipo. Cuando utilizamos %p como formato de salida en la función printf, estamos esperando un argumento que sea un puntero.



¿Qué paso entonces ?





¿Ahora programemos algo?

Hacer un programa que ingrese 2 valores **num1** y **num2** de tipo **entero** por teclado, los sume por medio de una función llamada **void Sumar** y guarde en una variable llamada **resul**, el resultado. Pero para ello tendrá que hacer lo siguiente, defina un puntero, el cual tendrá que contener el resultado donde se alojará la suma, por último, mostrar el resultado de la suma, por medio de ese puntero.



¿Ahora programemos algo?

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

```
// manejo de punteros
void sumar(int, int, int*);
```




¿Ahora programemos algo?

```
int main()
{
    int i, a, resultado=0;
    int *ptr;
    printf("Ingrese un valor de i:");
    scanf("%d", &i);
    printf("Ingrese un valor de a:");
    scanf("%d", &a);
    ptr=&resultado;
    sumar(i,a,ptr);
    printf("valor de la suma:%d\n",*ptr);
    printf("contenido de i=%d\n", i);
    printf("contenido de a=%d\n", a);
    printf("contenido de ptr=%d\n", *ptr);
    printf("direccion de a: %p\n", (void *)&a);
    printf("direccion de i: %p\n", (void *)&i);
    printf("direccion de ptr: %p\n", (void *)ptr);
    return 0;
}
```



¿Ahora programemos algo?

```
void sumar(int x, int y, int *r)
{
    *r=x+y;
}
```



2do Programa, ¿ Listos ?

Hacer un programa que ingrese 2 valores **num1** y **num2 de tipo entero** por teclado, El objetivo del programa es intercambiar los valores de las variables num1 y num2, desarrollando una función llamada intercambio, en ella se debe producir solo el intercambio, luego informar ese resultado.



2do Programa

```
int main()
{
    int num1, num2;
    printf("Ingrese el valor 1: ");
    scanf("%d", &num1);
    printf("Ingrese el valor 2: ");
    scanf("%d", &num2);
    printf("Valores originales: num1 = %d,
num2 = %d\n", num1, num2);
    // Llamando a la función intercambio
para intercambiar los valores
    intercambio(&num1, &num2);
    printf("Valores intercambiados: num1 =
%d, num2 = %d\n", num1, num2);
    return 0;
}
```

```
// Función para intercambiar dos valores
usando referencias
void intercambio(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
void intercambio(int *, int *);
```



3er Programa, ¿Vamos ?

Escribe una función que calcule la **potencia** de un número entero elevado a otro número entero utilizando punteros. Para ello solicite al usuario, el valor de la base y el valor del exponente.



3er Programa, ¿Vamos ?

```
void potencia(int *base, int *exponente, int *resultado) {  
    *resultado = 1;  
    for (int i = 0; i < *exponente; i++) {  
        *resultado *= *base;  
    }  
}
```



4to Programa, ¿Vamos ?

implementar una función en C que reciba un número entero y devuelva su valor absoluto utilizando punteros. Además, se proporcionará un programa principal que use esta función para calcular y mostrar el valor absoluto de un número dado.



¡¡5to Programa, Seguimos !!

Enunciado del Programa: Competencia de Recolección de Caracoles

Desarrollar un programa en ANSI C que simule una competencia de recolección de caracoles en la playa. El programa deberá cumplir con las siguientes especificaciones:

- 1.El usuario indicará cuántos grupos de personas participarán en la competencia, con un máximo de 3 grupos.
- 2.Cada grupo puede estar compuesto por hasta 3 personas.
- 3.La meta de cada grupo es recolectar la mayor cantidad de caracoles posible.
- 4.El programa deberá determinar y anunciar:
 1. El total de caracoles recolectados por todos los grupos.
 2. Cuál de los grupos recolectó la mayor cantidad de caracoles y, por lo tanto, ganó la competencia.



¡¡6to Programa, avanzando !!

Enunciado del Programa:

Escribir dos funciones que permitan calcular:

- La cantidad de segundos en un tiempo dado en horas, minutos y segundos.
- La cantidad de horas, minutos y segundos de un tiempo dado en segundos.

Escribe un programa principal con un menú donde se pueda elegir la opción de convertir a segundos, convertir a horas, minutos y segundos o salir del programa.



GAME OVER

Profesor JTP
MCP. Miguel Silva

Profesores Titulares
Ing. Israel Pavelek
Ing. Behringer Alejandro