

Ejercicios Punteros

Corrija los siguientes códigos e indique qué imprime.

Observación: “%p” se ocupa para ver la dirección de memoria.

1.- <pre>int *a, *b = NULL; if(a == null) printf("A es null"); if(b == NULL) printf("B es NULL");</pre>	2.- <pre>int *a = NULL, *b = NULL; if (a == b) printf("Iguales en referencia"); else printf("No iguales en referencia");</pre>
3.- <pre>int *a, *b = NULL; if(a == b) printf("Misma referencia entre A y B"); else printf("Referencias distintas entre A y B"); printf("\nA: %p - B: %p", a, b);</pre>	
4.- <pre>int a, *b = NULL; printf("A:"); scanf("%d", a); printf("B:"); scanf("%d", *b); printf("%d %d", a, b);</pre>	
5.- <pre>int *n, m = 10; m = &n; printf("%d %d", *n, m);</pre>	6.- <pre>int a = 1, s = 2, *x; *x = a + s; printf("%d", x);</pre>

<p>7.-</p> <pre>int x1 = 1, x2 = 2, x3; x3 = *x1 + *x2; printf("%d", *x3);</pre>	<p>8.-</p> <pre>int x1 = 1, *x2; x2 = x1; x2++; printf("%d - %d", *x1, *x2);</pre>
<p>9.-</p> <pre>int *z = 1, *x = 2; printf("(%p - %d) - (%p - %d)", *z, z, *x, x);</pre>	
<p>10.-</p> <pre>char t= 'A', *m; *m = &t; *m++; printf("%c - %c", t, *m);</pre>	<p>11.-</p> <pre>char a = 'a'; int *b = (int*)malloc(sizeof(int)); *b = a; a++; printf("%c %c", a, b);</pre>
<p>12.-</p> <pre>int a = 0, *b, *c; b = &a; c = &a; *b++; *c++; printf("(%p - %d) (%p - %d) (%p - %d)", a, *a, b, &b, *c, c);</pre>	

13.-

```
int *a, *b;

*a = (int*)malloc(sizeof(int));
*b = (int*)malloc(sizeof(int));

*a = *b = 2;

printf("(%p - %d) (%p - %d)", &a, a, &b, b);

a = b;

printf("\n(%p - %d) (%p - %d)", &a, a, &b, b);
```

14.-

```
int *a, b, c;

*a = (int*)malloc(sizeof(int));
a = b = c = 10;

printf("(%p - %d) (%p - %d) (%p - %d)", a, a, b, b, c, c);

a = &c;
b++;
c--;

printf("(%p - %d) (%p - %d) (%p - %d)", a, a, b, b, c, c);
```

Ejercicios funciones

1.- Implementar una función que calcule la potencia de un número.

Ejecute su función en el Main e imprima el resultado obtenido.

Restricción 1: Para todas las declaraciones de variables que se utilicen dentro de su función deberán ser punteros.

Restricción 2: Utilice el siguiente prototipo de función. (No es posible modificar)

Observación: En el argumento **int *r** debe guardar el resultado.

```
void potencia(int *a, int *b, int *r)
```

2.- Implementar una función que calcule la potencia de un número de **forma recursiva**.

Ejecute su función en el Main e imprima el resultado obtenido.

Restricción 1: Para todas las declaraciones de variables que se utilicen dentro de su función deberán ser punteros.

Restricción 2: Utilice el siguiente prototipo de función. (No es posible modificar)

Observación 1: El argumento **int *r** siempre debe iniciar con el valor de 1.

Observación 2: En el argumento **int *r** debe guardar el resultado.

```
void potencia_r(int *a, int *b, int *r)
```

3.- Implementar una función que calcule el factorial de un número.

Restricción 1: Para todas las declaraciones de variables que se utilicen dentro de su función deberán ser punteros.

Restricción 2: Utilice el siguiente prototipo de función. (No es posible modificar)

```
void factorial(int *n, int *r)
```