

# Problemas de aritmética de punteros

**PROBLEMA 1:** Encuentre los errores en cada una de las líneas suponiendo las siguientes declaraciones de variables: `int *pta, *ptb, a, b`.

Descripción	Código	Error
pta apunta a la variable a	<code>pta = *a;</code>	
ptb apunta a donde apunta pta	<code>ptb = *pta;</code>	
Contenido de pta igual a 8	<code>pta = 8;</code>	
Aumentar en 3 unidades el contenido de ptb	<code>ptb = ptb + 3</code>	
ptb apunta a NULL	<code>ptb = &amp;NULL;</code>	
Almacenar 8 en b	<code>&amp;b = 8;</code>	
ptb apunta a b	<code>*ptb = b;</code>	
Contenido de ptb es a + 10	<code>ptb = a + 10;</code>	
Imprimir la suma de los contenidos de pta y ptb	<code>printf("%d", pta + ptb);</code>	

**PROBLEMA 2:** ¿Qué contienen cada una de las variables después de ejecutarse el siguiente fragmento de código?

```
int *ptc, *ptn, b = 20;
ptc = &b;
ptn = (int*)malloc(sizeof(int));
*ptn = b + 15;
*ptc = *ptc - 1;
ptc = ptn;
*ptc = 2;
```

Variable	Valor
ptc	
ptn	
b	

**PROBLEMA 3:** Descubra los errores:

Código erróneo	Código correcto
<pre>char *pta, car1; int *ptc, b;  *pta = car1; b = (int*)malloc(sizeof(int)); *b = 8; ptc = (int)malloc(sizeof(int)); ptc = b + 5; free(*ptc);</pre>	

**PROBLEMA 4:** Modifique el código aplicando aritmética de punteros:

Código a modificar	Código modificado
<pre>int AE[4] = {4, 1, 2, 5}, *p, i;  p = AE; for(i = 0; i &lt; 4; i++)     p[i] = p[i] + 3; p = (int*)malloc(9*sizeof(int)); for(i = 0; i &lt; 9; i++)     p[i] = 5;</pre>	

**PROBLEMA 5:** Realice las siguientes instrucciones:

Descripción	Código
Declare dos punteros a carácter A y B Asígnele memoria a B para 20 caracteres	
Copie en B la palabra “informática” Utilizando aritmética de punteros, pase todos los caracteres de B a mayúsculas	
Haga que A apunte al carácter que se encuentra en la mitad de la cadena B mediante aritmética de punteros.	

**PROBLEMA 6:** Declare una variable cadena de 100 caracteres. Suponiendo que esta cadena almacena un número indeterminado de caracteres (menos de 100) seguido de un carácter nulo ('\0'). Escriba un fragmento de código que localice la posición del primer carácter espacio (' ') usando aritmética de punteros.

**PROBLEMA 7:** Declare una variable números de 1000 enteros. Escriba un fragmento de código que recorra el vector y modifique todos los enteros negativos cambiándolos de signo. No se debe usar el operador '[]', es decir, se debe usar aritmética de punteros.

**PROBLEMA 8:** ¿Qué imprime el siguiente código?

Código	Resultado
<pre>#include &lt;stdio.h&gt; void func1(int u, int v); void func2(int *pu, int *pv); main () {     int u=1;</pre>	

```
int v=3;
printf("Antes de func1:
      u=%d, v=%d\n", u, v);
func1(u, v);
printf("Después de func1:
      u=%d, v=%d\n", u, v);
printf("Antes de func2:
      u=%d, v=%d\n", u, v);
func2(&u, &v);
printf("Después de func2:
      u=%d, v=%d\n", u, v);
system("PAUSE");
}

void func1(int u, int v)
{
    u=0;
    v=0;
    printf("Dentro de func1:
          u=%d, v=%d\n", u, v);
}

void func2(int *pu, int *pv)
{
    *pu=0;
    *pv=0;
    printf("Dentro de func2:
          *pu=%d, *pv=%d\n", *pu, *pv);
}
```

**PROBLEMA 9:** Dibuje el estado de la memoria tras ejecutarse el siguiente fragmento de código:

```
int *ptr1, *ptr2;
int a = 15, b[5] = {1, 2, 3,
4, 5};
ptr1 = &a;
ptr2 = &b[0];
*(ptr2+2) = a;
*ptr1 = b[4];
ptr1 = ptr2 + 3;
```

**PROBLEMA 10:** Indique a qué equivalen las siguientes expresiones suponiendo las siguientes declaraciones: `int a[5] = {9, 8, 7, 6, 5}`, `b[3] = {1, 2, 3}`

Expresión	Equivalencia
<code>b[2]</code>	
<code>a[1]</code>	
<code>*(b+1)</code>	
<code>a[4]</code>	
<code>*(a+3)</code>	

**PROBLEMA 11:** Indique el valor de las siguientes variables después de ejecutarse el fragmento de código que se muestra a continuación:

```
char *ptr1, *ptr2, c = 'm';
int *ptm, *ptn, n = 5;

ptr1 = &c;
ptm = (int*)malloc(sizeof(int));
*ptr1 = c + 2;
*ptm = 3;
ptr2 = ptr1;
ptn = &n;
*ptn = *ptm + *ptn;
*ptr2 = 'k';
ptm = ptn;
*ptn = n * 3;
```

Variable	Valor
<code>ptr1</code>	
<code>ptr2</code>	
<code>c</code>	
<code>ptm</code>	
<code>ptn</code>	
<code>n</code>	

**PROBLEMA 12:** Muestre el fragmento de código correspondiente a la inicialización de los elementos de una matriz mediante aritmética de punteros. La matriz será declarada como un puntero a un puntero a entero. El tamaño de la matriz es de  $5 \times 4$  y se inicializará a 2.