

GUÍA PRÁCTICA DE PUNTEROS

Resolvé los ejercicios utilizando el lenguaje C o pruebas de escritorio, según corresponda. Asegurate de leer al menos dos veces los enunciados antes de intentar confeccionar las soluciones.

ENUNCIADOS

Operaciones con punteros

1) Realizá un programa en C que resuelva cada una de las siguientes indicaciones:

- A) Declarar las variables enteras **a** con valor **10** y **b** con valor **20**.
- B) Declarar **p** y **q** como punteros a **int**.
- C) Asignar a **p** la dirección de **a**.
- D) Imprimir el valor de **a** y su dirección.
- E) Imprimir la dirección contenida en **p**, el valor al que apunta **p** y la dirección de **p**.
- F) Asignar a **q** la dirección de **a**.
- G) Imprimir la dirección contenida en **q**, el valor al que apunta **q** y la dirección de **q**.
- H) Sumarle 1 a la variable **a** usando el puntero **p**.
- I) Ejecutar nuevamente los puntos E) y G)
- J) Copiar y ejecutar estas líneas:

```
if (p == q) printf("P y Q contienen el mismo valor\n");  
else printf("P y Q contienen distinto valor\n");  
  
if (*p == *q) printf("Los contenidos apuntados desde P y Q son iguales\n");  
else printf("Los contenidos apuntados desde P y Q son distintos\n");
```

- K) Asignar a **q** la dirección de **b**.
- L) Modificar **b**, mediante **q**, haciendo que valga **22**.
- M) Ejecutar nuevamente el punto J)
- N) Ejecutar nuevamente el punto I)

2) Realizá el seguimiento del siguiente código mediante una prueba de escritorio e indicá cuál es su salida:

```
#include <stdio.h>  
  
int main() {  
    int a = 1, b = 3;  
    int c, *pa, *pb;  
  
    pa = &a;  
    *pa = 2 * a;  
    pb = &b;
```

```
c = 3 * (*pb - *pa);

printf("a=%d, b=%d, c=%d, *pa=%d, *pb=%d\n", a, b, c, *pa, *pb);

return 0;
}
```

3) Identificá el/los error/es del siguiente programa:

```
#include <stdio.h>

int main() {
    int x = 5;
    float y = 5;

    int *xPtr = NULL;
    printf ("%d", *xPtr);

    xPtr = &y;
    printf ("%d", *xPtr);

    return 0;
}
```

4) Realizá el seguimiento del siguiente código mediante una prueba de escritorio e indicá cuál es su salida:

```
#include <stdio.h>

int main() {
    int x = 5;

    int *xx = &x;
    *xx = *xx * x;
    printf ("x vale %d y *xx vale %d", x, *xx);

    return 0;
}
```

5) Realizá el seguimiento del siguiente código mediante una prueba de escritorio e indicá cuál es su salida:

```
#include <stdio.h>

int main() {
    int a, b, *p, *q, *r;
    char c, d, *m, *n;
```

```

b = 7;
a = 5;
q = &a;
p = q;
*p = b;

printf("*Q=%d\n\n", *q);

r = &b;
*r = *q;
printf("A=%d, b=%d, *p=%d, *q=%d, *r=%d\n\n", a, b, *p, *q, *r);

c = 'A';
d = 'B';
m = &c;
n = &d;
*m = *n;

if (m == n) printf("M y N apuntan a la misma direccion\n");
else printf("M y N apuntan a direcciones distintas\n");

printf("*M=%d\n", *m);
printf("\n");

return 0;
}

```

6) Ejecutá el siguiente código y deducí qué significa el **:

```

#include <stdio.h>

int main() {
    int *p, a=1, b=8, **s;

    p=&a;
    b=*p+1;
    s=&p;
    printf("A=%d &A=%d\n\n", a, &a);
    printf("B=%d &B=%d\n\n", b, &b);
    printf("P=%d *P=%d &P=%d\n\n", p, *p, &p);
    printf("S=%d *S=%d &S=%d\n\n", s, *s, &s);

    *s=&b;
    printf("Se ejecuto *S=&B\ny ahora\n\n");
}

```

```
printf("S=%d *S=%d &S=%d\n\n", s, *s, &s);

**s=79;
printf("A=%d &A=%d\n\n", a, &a);
printf("B=%d &B=%d\n\n", b, &b);
printf("P=%d *P=%d &P=%d\n\n", p, *p, &p);

printf("\n");
return 0;
}
```

7) Realizá un programa en C que resuelva cada una de las siguientes indicaciones:

- A) Declará **a** y **b** como **int**. **a** debe valer **10** y **b** debe valer **20**.
- B) Declará **c** y **d** como **char**. **c** debe valer **'A'** y **d** debe valer **'E'**.
- C) Declará **p** como puntero a **int**. Asigne la dirección de **a**.
- D) Modificá el contenido de **a** usando **p**, asignando el valor **64**.
- E) Declará **q** como puntero a **char**. Asigne la dirección de **c**.
- F) Modificá el contenido de **c**, asignándole el de **d**, usando **q**.
- G) Imprimí los contenidos de **p**, **&p**, ***p**, **q**, **&q**, ***q**, **a**, **b**, **c** y **d**.
- H) Asigná a **q** (previo casting) el valor de **p**.
- I) Ejecutá nuevamente el punto **G**.
- J) Sumale **2** a ***p**.
- K) Ejecutá nuevamente el punto **G**.
- L) Declará **pp** como puntero a puntero a **int**. Asigne la dirección de **p**.
- M) Hacé que **p** apunte a **b** usando **pp**.
- N) Imprimí los contenidos de **p**, ***p**, **pp**, ***pp**, ****pp**.

8) Teniendo una variable entera **int x = 88**; y declarando los siguientes punteros:

```
int *ptr1, **ptr2, ***ptr3, ****ptr4;
```

Construí un programa en C que permita imprimir 4 veces el valor de la variable **x (88)** utilizando cada uno de los 4 punteros declarados.

Pasaje por referencia

9) Creá la variable entera **x** en **main** con valor **10**. Luego, pasala como argumento de la función **duplicar**, la cual debés definir para recibir un valor entero y duplicar su valor por referencia. Mostrá finalmente, en el **main**, el valor de **x** antes y después de la invocación de **duplicar**.

10) Definí y probá desde el **main** la función **leerColorRGB** que reciba tres referencias a enteros por parámetro, que representan los valores de cada canal (**Red**, **Green**, **Blue**). La función debe leer de la consola un valor entero (entre **0** y **255**) por cada canal y asignarlo. Mostrar finalmente, en el **main**, los valores de cada canal tras la invocación de **leerColorRGB**.