



Programación

Guía de Ejercicios de punteros

1. Considere las siguientes instrucciones:

```
int *p;  
int i;  
int k;  
i = 42;  
k = i;  
p = &i;
```

Luego de esas instrucciones, ¿cuál de las siguientes cambia el valor de i a 75?

- A. `k = 75;`
- B. `*k = 75;`
- C. `p = 75;`
- D. `*p = 75;`
- E. Dos o más de las anteriores.

2. Explicar qué está mal en el siguiente fragmento de código:

```
char c = 'A';  
double *p = &c;
```

3. Escribir un programa que imprima los elementos de un vector de enteros en orden inverso utilizando punteros (no usar subíndices []).

```
int v[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
int *p;  
  
// Escribir el código aca
```

4. Escribir una función que reciba un vector de enteros y su tamaño, y devuelva la cantidad de números impares que contiene.

```
int impares (int *v, int tam);
```

5. Escribir una función que devuelva un puntero al máximo valor de un vector de "doubles". Si el vector está vacío (cant = 0) la función debe retornar NULL.

```
double* max (double *v, int cant);
```

6. Suponga que se ejecutan las siguientes instrucciones:

```
int a[] = {5, 15, 34, 54, 14, 2, 52, 72};  
int *p = &a [1], *q = &a [5];
```



- a. ¿Cuál es el valor de $*(p+3)$?
- b. ¿Cuál es el valor de $*(q-3)$?
- c. ¿Cuál es el valor de $q-p$?
- d. ¿La condición $p < q$ es verdadera o falsa?
- e. ¿La condición $*p < *q$ es verdadera o falsa?

7. Suponga que **primero**, **ultimo** y **medio** son punteros del mismo tipo y **primero** y **ultimo** apuntan al primer y último elemento de un vector. Si quiere que **medio** apunte al elemento que está en la mitad del vector ¿por qué la siguiente instrucción es inválida y cómo se podría solucionar?

```
medio = (primero + ultimo) / 2;
```

8. Implemente una función que reciba un vector de números en punto flotante y un entero que representa el tamaño del vector y devuelva el promedio.

Usar notación aritmética de punteros (no subíndices []) para recorrer el vector.

9. Implemente la siguiente función para calcular el producto escalar de dos vectores:

```
double prod_esc (double *a, double *b, int n);
```

Donde a y b apuntan a vectores de longitud n.

La función debe retornar $a[0]*b[0]+a[1]*b[1]+ \dots + a[n-1]*b[n-1]$

Usar notación aritmética de punteros (no subíndices []) para recorrer los vectores.

10. Escriba un programa que permita al usuario ingresar valores enteros en una matriz de 3x3 y luego imprima los elementos de la diagonal principal.

Usar notación aritmética de punteros (no subíndices []) para recorrer la matriz.

11. Suponga que el puntero p está declarado como sigue:

```
struct mi_estructura {  
    char a, b;  
    int c;  
    int e [5];  
};  
  
struct mi_estructura me;  
struct mi_estructura *p = &me;
```



¿Cuáles de las siguientes instrucciones son válidas?

- a) `p->b = ' ';`
- b) `p->e [3] = 10;`
- c) `(*p).a = '*';`
- d) `*p.b = '?';`
- e) `me.c = 1;`
- f) `p.c = 1;`

12. Suponga que `f` y `p` están declarados como sigue:

```
struct {  
    union {  
        char a, b;  
        int c;  
    } d;  
    int e [5];  
} f, *p = &f;
```

¿Cuáles de las siguientes instrucciones son válidas?

- g) `p->b = ' ';`
- h) `p->e [3] = 10;`
- i) `(*p).d.a = '*';`
- j) `p->d->c = 'x';`