



Ejercicio teórico-práctico sobre punteros

Objetivos:

Que los alumnos aprendan y practiquen los conceptos de:

- Enumeración
- Múltiples fuentes
- Punteros

Para esto, deben haber visto el video de la clase, y leído toda la documentación pertinente de la unidad.



Formato de entrega:

Para la correcta visualización de los ejercicios por parte de los docentes, es importante que:

- Creen un proyecto para cada ejercicio a resolver, respetando la numeración para la nomenclatura.
- No agregar ni quitar archivos dentro de la carpeta del proyecto.
- Todos los proyectos deben guardarse en una misma carpeta, con el siguiente nombre:

“Apellido-Nombre-NumeroEjercicioSemanal”.

Dicha carpeta deben comprimirla en formato .RAR o .ZIP

Adjuntar el comprimido en la devolución de la tarea.

Consideraciones previas:

Realizar todos los ejercicios en un mismo proyecto, y enviar un único proyecto al aula virtual. Separar todos los enunciados por una línea de comentario. Ejemplo: Una vez hecho el ejercicio 1, se recomienda comentar el código antes de comenzar con el ejercicio 2, para evitar que sea parte del programa al compilar. A modo de ejemplo:

```
Int main ()
```

```
{
```

```
/*
```

```
CODIGO DE EJERCICIO 1
```

```
*/
```

```
Comienza ejercicio 2
```

```
Return 0;
```

```
}
```

Enunciado:

Se pide armar un programa que ofrezca como menú de opciones, la resolución de cada ejercicio dictado a continuación. Para ello, deberán utilizar las herramientas de la estructura “do-while” y “switch” para hacer un menú repetitivo, hasta que se ingrese la opción de salida.

Para ello contarán un código semicompleto donde deberán ustedes agregar lo que falte donde se indique que hay que completar.

1. Ejercicio 1: Punteros, ¿Qué imprime? Calcular primero mentalmente, luego verificar con código.

```
int *punt;  
int x=7;  
int y=5;  
punt=&x;  
*punt=4;  
printf(“%d, %d”,x,y); // ¿qué imprime este printf?
```

2. Ejercicio 2: Punteros, ¿Qué imprime? Calcular primero mentalmente, luego verificar con código.

```
int *punt;  
int x=7;  
int y=5;  
punt=&x;  
x=4;  
printf(“%d, %d”, *punt,y); // ¿qué imprime este printf?
```

3. Ejercicio 3: Punteros, ¿Qué imprime? Calcular primero mentalmente, luego verificar con código.

```
int *punta, *puntb;  
int x=7;  
int y=5;  
punta=&x;  
*punta=3;  
puntb=&y;  
*puntb=x;  
x=9;  
puntb=punta;  
printf(“%d, %d”, *puntb, y); // ¿qué imprime?
```

4. Ejercicio 4: Punteros y arrays, ¿Qué imprime? Calcular primero mentalmente, luego verificar con código.

```
int *punt,i;  
int x[5]={1,2,3,4,5};  
punt=x;  
*punt=9;  
for(i=0;i<5;i++)  
printf("%d,",x[i]); // ¿qué imprime?
```

5. Ejercicio 5: Punteros y arrays, ¿Qué imprime? Calcular primero mentalmente, luego verificar con código.

```
int *punt,i;  
int x[5]={1,2,3,4,5};  
punt=x;  
*(punt+2)=9;  
*(x+3)=7;  
punt[1]=11;  
for(i=0;i<5;i++)  
printf("%d,",*(punt+i)); // ¿qué imprime?
```