

Ejercicios de repaso

Qué imprime?

```
float i;  
for (i = 1.28; i < 16; i*=4)  
    printf("%.1f ", i);
```

Qué imprime?

```
int i, a=0;  
for (i = 0 ; i==100 ; i++)  
    a += 3;  
printf("el valor de a es %d ",a);
```

- a. El valor de a es 300.
- b. El valor de a es 0.
- c. El valor de a es 100.
- d. El valor de a es 303.
- e. No imprime nada dado que el código presenta errores en la sintaxis del *for* al compilar.
- f. No imprime nada dado que queda en un bucle infinito.

Qué imprime?

```
int i;  
for (i = 5; i > 0; i-=2)  
    printf("%d ", !(i % 3));
```

Qué imprime?

```
int i;  
for (i = 5; i > 0; i=-2)  
    printf("%d ", !(i % 3));
```

Qué imprime?

```
int i;  
for (i = 1; i < 4; i++){  
    if (i==2)  
        continue;  
    printf("%d ", i);  
}
```

Qué imprime?

```
int i, a;  
for (i = 9; i ; i/=3) {  
    a = i-3 ? ++i : i++ ;  
    printf("%d ", a );  
}
```

Qué imprime?

```
int i;  
for (i=0 ; i<3 ; ++i) {  
    static int s = 4;  
    if (--s % 3)  
        printf("s = %d\n", s--);  
}
```


Indique la opción verdadera

- a. Un programa puede compilar con *errores* pero no con *warnings*.
- b. Un programa puede compilar con *warnings* y *errores*.
- c. Un programa puede compilar con *warnings* pero no puede ejecutarse.
- d. Un programa con *warnings* puede ejecutarse pero podrían aparecer resultados inesperados.
- e. Los *warnings* son errores críticos.

Para cada inciso indique si es verdadero (V) o falso (F)

	No es posible asignar ningún valor entero a una variable de tipo puntero.
	No es posible comparar dos variables estructuras aunque sean del mismo tipo.
	Puede utilizarse la función <code>realloc()</code> para realizar la misma operación que la función <code>malloc()</code> .

Para cada inciso indique si es verdadero (V) o falso (F)

	Si una función recibe como parámetro <code>int * const Ptr</code> no podrá modificar lo apuntado por <code>Ptr</code> .
	Si se imprime una variable <code>char</code> (toma valores entre -128 y 127) inmediatamente después de asignarle el valor 128 se visualizará el valor -128.
	Un puntero <code>void</code> puede asignarse a cualquier tipo de puntero.

Utilizando la sigte definición

```
struct alu {  
    char apellido[50];  
    char nombre[50];  
    char legajo[8];  
};
```

- a) Renombre el tipo **struct alu** a **alumno**.
- b) Defina una función que permita inicializar una estructura **alumno**.
- c) Defina un arreglo de 10 elementos de tipo alumno e inicialice cada uno de ellos utilizando la función definida en el punto b).
- d) Imprima la información de cada alumno con el siguiente formato:
Apellido y nombre: Perez, Juan | Legajo: 7751/8
Apellido y nombre: García, Pablo | Legajo: 6952/1

Ejercicio de Matrices y mem.dinámica

- Descargue el archivo “Matrices y memoria dinámica.pdf” y analice el código C que contiene.
- indique cuales son las líneas sintácticamente incorrectas y suprímalas del código.
- Luego escriba el código de la función **sumaUltimaFila()** que dada una matriz y la cantidad de filas y columnas que posee retorna la suma de los elementos de la última fila.
- Finalmente indique que imprime.