

Apellido y Nombre:										
Legajo:		Hojas:		Archivos:						
UTN-FRBA	Informática I	R1051	1er Parcial		10-Ago-2015					

## Parte Teórica

Considere el siguiente bloque de código

```
int arr[4] = {2500, 3500, 4500, 5500};
int *ptr1, *ptr2;
ptr1 = arr;
ptr2 = &arr[2];
```

Escriba la salida del programa al lado de cada línea de código. Las instrucciones se ejecutan en secuencia. Justifique cada una de las respuestas. Asuma que `arr` esta ubicado en `0x7fff0000`, `ptr1` en `0x7fff1000` y `ptr2` en `0x7fff2000`.

1) <code>printf ("%ld \n", (ptr2 - ptr1));</code>	_____
2) <code>printf ("%ld \n", ((long) ptr2 - (long) ptr1));</code>	_____
3) <code>printf ("%d \n", (*ptr2 - *ptr1));</code>	_____
4) <code>printf ("%p \n", ptr2);</code>	_____
5) <code>printf ("%x \n", *ptr2);</code>	_____
6) <code>printf ("%p \n", ptr2++);</code>	_____
7) <code>printf ("%x \n", *(ptr2++));</code>	_____
8) <code>printf ("%p \n", ptr2);</code>	_____

## Parte Práctica

1. Como parte de un programa para validar strings se pide implementar cuatro funciones que retornen verdadero ó falso para los siguientes casos:

- a) `int validar_email(char *)`; Valida direcciones de email del tipo `usuario@dominio.com`
- b) `int validar_ip(char *)`; Valida direcciones IP del tipo `0.0.0.0` a `255.255.255.255`
- c) `int validar_tarjeta(char *)`; Valida números de tarjetas de crédito del tipo `1111-2222-3333-4444`
- d) `int validar_fecha(char *)`; Valida números de fechas con formato `AAAA-MM-DD#HH:MM:SS`

Estas funciones deberán estar en un archivo `fvalidar.c` y su correspondiente archivo header `fvalidar.h`. Se pide realizar todas las validaciones posibles para asegurar que el string pasado como argumento es un string válido.

2. Realizar un programa de validación que reciba por linea de comando el tipo de validación a realizar y a continuación todos los strings a validar. Por ejemplo:

```
./validar email juan@gmail.com otro@yahoo.com cualquier_cosa
```

cuya salida sea:

```
juan@gmail.com => email valido
otro@yahoo.com => email valido
cualquier_cosa => email invalido
```

Los tipos de validación permitidos son: `email`, `ip`, `tarjeta`, `fecha`. El programa debe validar éstos tipo así como la cantidad mínima de parámetros a pasar por la linea de comandos. El código fuente del programa estará en un archivo `validar.c` y hará uso de las funciones anteriormente escritas. Se pueden utilizar todas las funciones de la librería estándar que crea conveniente.

Se recomienda la utilización de archivos de datos para cada uno de los tipos de validación para facilitar las pruebas, así como un script para realizar la compilación y el linkeo. Si no se usa un script de compilación indique cómo compilar y linkear los archivos para obtener el ejecutable.

Finalmente, complete la cantidad de hojas y archivos que entrega en ésta hoja.