

- 1-) `int a = 33;`  
`int *punto;`  
`punto = &a;`
- 2-) Dir de memoria de a por ejemplo = FF8.  
 y el %d = 10. El %p se usa para mostrar la posición de memoria
- 3-) Muestra el valor de número porque `punto` es precedido por `*` que el operador de dirección que muestra el contenido de la dirección a la que apunta `punto`.

### Ejercicios Diagnósticos Punteros

- 1) `int a=3; *punto; punto = a;` este código es correcto?, de ser falso, hacerlo correcto.
- 2) `int a; a=10; printf("Dirección de a = %p, valor de a = %i\n", &a, a);` porque el %p? que muestra cada cosa?
- 3) `int numero; int *punto; numero = 43; punto = &numero; printf("Dirección de numero = %p, valor de numero = %i\n", &numero, *punto);` porque \*punto muestra el valor de numero?
- 4) `int a, b, *punto1, *punto2; a = 5; b = 5; punto1 = &a; punto2 = &b; if ( *punto1 == *punto2 ) printf("Son iguales\n");` que camino toma el if? Que debemos cambiar en el if para que sea true?
- 5) A partir del siguiente prototipo  $\rightarrow$  `char * compara (char *, char [7][20])`, la función `compara` recibe como parámetro el puntero del comienzo de la palabra ingresada en el main y el segundo parámetro es un arreglo bidimensional que se debe cargar en el main, el cual tiene los días de semana, la función debe devolver "día escrito correctamente" o "día escrito incorrectamente".
- 6) A partir del siguiente prototipo `char * carga(void)`; cargar un arreglo de caracteres en la función y devolver un puntero a la posición de memoria del comienzo del mismo.
- 7) `void carga(char*, int*)`; el primer parámetro deja disponible en el main el texto ingresado y en el segundo la cantidad de caracteres que contiene. Mostrar luego de la llamada la cadena mediante la estructura `for`.
- 8) Se desea formar una tabla de cadena de caracteres, NO REPETIDAS. En dicha tabla debe figurar cada cadena ingresada, la cantidad de caracteres alfabéticos de la misma. Se admitirá un máximo de 100 cadenas y cada una hasta 25 caracteres.
- 9) Se desea formar una tabla de cadena de caracteres, NO REPETIDAS. En dicha tabla debe figurar cada cadena ingresada, la cantidad de caracteres alfabéticos de la misma. Se admitirá un máximo de 100 cadenas y cada una hasta 25 caracteres.

`char * Ingresa_Cadena ( )` NO PROGRAMAR

`struct tab* Forma_Tabla (char *, int *filas)` PROGRAMAR

La función `Forma_Tabla`, recibe la dirección de la cadena, la almacena en la tabla si NO está ya almacenada y comparte con su llamada la cantidad de filas ocupadas de la misma a través del parámetro `int *filas`. Calcula y almacena la cantidad de caracteres alfabéticos de la cadena correspondiente.

```
struct tab
{ char cadena[25];
  int alfa;} // la cantidad de caracteres alfabéticos de
cadena
```

El main debe generarse de esta manera: `do {`

```
  system("cls");
  cadena = Ingresa_Cadena();
  fflush(stdin);
  p_tabla = Forma_Tabla (cadena, &filas);
  printf ("\n\nDesea continuar ingresando palabras? (s/n) ");
  scanf ("%c", &rta);
} while (rta == 's');
```