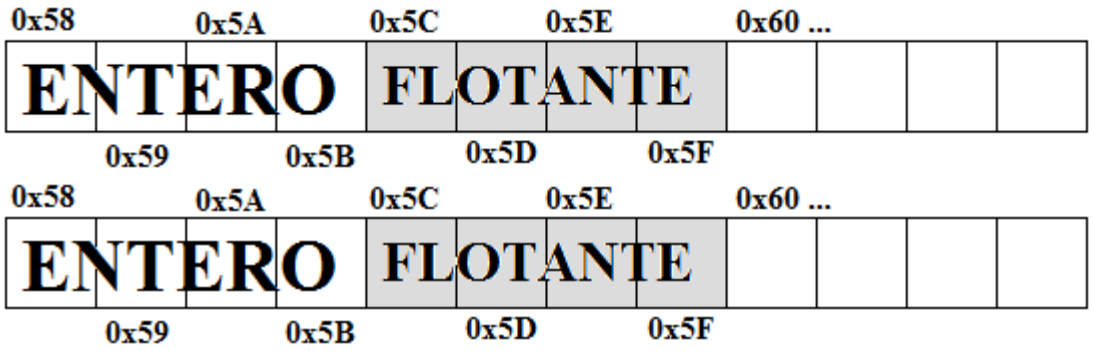
Apuntadores y memoria RAM (repaso)

Responda por equipo en una hoja del cuaderno para entregar a las siguientes preguntas:

1.- Declare en un programa en lenguaje C las siguientes variables e imprima las direcciones de memoria en las cuales se encuentra almacenada cada una de ellas (para imprimir direcciones utilice el especificador de formato `%p`). Toda dirección se escribe siempre en hexadecimal y para indicarlo se antepone el símbolo `0x`.

```
char caracter = 0;
int entero = 0;
float flotante = 0;
long largo = 0;
double doble = 0;
```

A partir de la impresión y con ayuda de la función `sizeof()` (véase con el comando `man`) dibuje en una hoja un mapa de memoria donde cada cuadro es un espacio de almacenamiento para un byte, junto con las direcciones y variables almacenadas, como muestra el siguiente ejemplo(por error no se encuentra dibujada la variable `char`).



2.- ¿Cuántos bits se almacenan en una dirección de memoria? ¿Cuál es el máximo valor hexadecimal, decimal y binario que puede almacenarse en una dirección de memoria? ¿Las variables se encuentran almacenadas en el orden en que fueron declaradas? ¿Las variables se almacenan en direcciones contiguas? ¿Los datos del tipo `float`, `long` y `double` ocupan la misma cantidad de bytes en todas las computadoras de su equipo?

3.- Observe de cuantos dígitos hexadecimales se componen sus direcciones de memoria al imprimirse. ¿Cuál sería la máxima dirección de memoria que se podría imprimir? (sugerencia: obtenga el tamaño de un apuntador) Si cada dirección de memoria corresponde al lugar en RAM donde se almacena un byte de información, ¿Cuantos bytes sería posible almacenar en la RAM (escriba el prefijo k, M o T según corresponda)?

4.- En realidad para almacenar direcciones de memoria se utiliza un conocidísimo tipo de variable denominada apuntador. Añada a continuación de las variables indicadas en el inciso uno, los siguientes apuntadores:

```
char *pcaracter;
int *pentero;
float *pflotante;
long *plargo;
double *pdoble;
```

Estas variables (tipo apuntador) también ocupan memoria. ¿Cuántos bytes ocupa cada apuntador?

Como es de esperarse los valores contenidos en estas variables de tipo apuntador tienen al inicio basura, compruébelo al imprimir su contenido. Posteriormente inicialice el apuntador `pcaracter` con la dirección de la variable `caracter`, compruebe que la asignación se ha hecho correctamente. ¿cómo lo comprobaría?

Finalmente agregue al dibujo realizado en el inciso uno, las variables tipo apuntador de este inciso.

5.- Ahora declare el siguiente arreglo

```
char cadena[] = "ESCOM - IPN";
```

Mediante un ciclo `for`, imprima la dirección de cada uno de los caracteres que conforman dicha cadena. ¿Son direcciones continuas o discontinuas? ¿Que caracter contiene la dirección cadena+4? ¿Cómo demostraría en su programa que después del último caracter 'N' efectivamente se encuentra el caracter fin de cadena? (sugerencia, véase el manual del código ascii con

`man ascii`).

6.- Inicialice una variable entera con el valor: 1234567890, posteriormente imprima el contenido de la variable en decimal y hexadecimal (utilice el especificador de formato `%d` y `%x`). ¿Corresponde el valor decimal con su equivalente en hexadecimal? Ahora repita la operación, pero inicialice la variable con el valor: -1234567890.

Para explicar porque no coincide en el caso de números negativos realice lo siguiente:

Pruebe a ver como se almacenan en RAM los siguientes valores: 0, 1, -1, 2, -2, 3, -3 …

Revise en Wikipedia el tema Representación de números con signo, y determine en su computadora como se almacenan los números con signo, haciendo conversiones a binario de los números hexadecimales (utilice la calculadora de UBUNTU en modo programador). Si ha entendido lo anterior responda ¿cuál es el máximo valor negativo y positivo que puede almacenar un entero en su computadora?, compruébelo asignando dichos valores a una variable.