

Los archivos generados deben respetar el siguiente formato de nombre `guiaDeClase06_ejercicioNumero.c`. Por ejemplo el archivo del ejercicio 1 debe llevar el nombre `guiaDeClase06_01.c`. Si el ejercicio tuviera ítems a y b por ejemplo el nombre deberá ser `guiaDeClase06_01_A.c` para el punto A.

Todos los archivos deberán ser subidos al repositorio dentro de una carpeta con el nombre `guiaDeClase06`. Todos los archivos deben estar comentados con `doxygen` y las respuestas a las preguntas realizadas deben ser contestadas usando el tag `\note`.

1. Implemente programa donde se instancie una variable tipo `int` y un puntero del mismo tipo. Apunte el puntero a esta variable e inicialícela usando el puntero con el valor `0x55`. Luego imprima el valor almacenado en la variable utilizando el puntero.
2. Escriba un programa que defina un vector de 10 elementos de tipo `unsigned char` inicialícelo con los números del 0 al 9 utilizando un puntero. Luego imprima dicho vector también usando un puntero.
3. Escriba un programa que defina un vector de 100 elementos de tipo `unsigned char`, además cree un puntero tipo `unsigned char` y otro `unsigned int`. Apunte ambos punteros al inicio del vector e increméntelos usando el operador `++`. Observe las direcciones que van tomando dichos punteros y obtenga conclusiones. ¿Cambiaría en algo el comportamiento de los punteros si el vector fuera `unsigned int`?
4. Implemente un programa que utilizando un puntero de tipo `unsigned char` imprima byte a byte una variable de tipo `int` en hexadecimal que ha sido inicializada con el número `0x12345678`. ¿El byte menos significativo del `int` en qué dirección se encuentra respecto al byte más significativo?
5. Implemente un programa en el cual se defina un vector de 4 `unsigned char`, inicialícelo con los siguientes elementos `0xAA`; `0xBB`; `0xCC`; `0xDD`, apúntelo con un puntero de tipo `int` e imprima por `stdout` el contenido de dicho puntero en hexadecimal.
6. Implemente una función que calcule el promedio de un vector de tipo `float`. El prototipo de la función es `float promedio (float *dataPtr, unsigned int dataCant);`
Donde: `dataPtr`: Es el puntero a los datos.
`dataCant`: Es la cantidad de elementos del vector apuntado.
7. Implemente una función que invierta el contenido de dos variables cuyo prototipo es:
`void swap (int *a, int *b);`
Donde: `a` y `b` son los punteros a las variables que se les debe invertir el contenido.
8. Implemente una función que verifique si un vector de `int` está ordenado de manera creciente o decreciente. El prototipo es `int orden (int *dataPtr, int dataCant);`
Donde: `dataPtr`: Es el puntero a los datos
`dataCant`: Es la cantidad de elementos del vector apuntado
Devuelve: 1 si el orden es creciente; 0 si no está ordenado; -1 si el orden es decreciente.

9. Implemente una función con el siguiente prototipo que devuelva las raíces de una ecuación de segundo grado. `int ecu2 (float *coef, float *r0, float *r1)`
Donde: coef: puntero al vector de coeficientes. (3 elementos)
r0, r1: Puntero a un vector de dos elementos que contiene la parte real e imaginaria de cada raíz. La función devuelve 1 si los coeficientes corresponden a una ecuación de segundo grado; cero en caso contrario y coloque NAN en la parte real e imaginaria de r0 y r1.
10. Implemente una función que devuelva un puntero al elemento que contiene el máximo valor de un vector. `int * myMax (int *dataPtr, unsigned int dataCant);`
Donde: dataPtr: Es el puntero a los datos
dataCant: Es la cantidad de elementos del vector apuntado
Devuelve: El puntero al elemento que contiene el máximo
11. Implemente una función que imprima en hexadecimal todos los caracteres de un string. El prototipo de la función es el siguiente. `int myHexa (char *dataPtr)`
Donde: dataPtr: Es el puntero al string a pasar a hexadecimal.
Devuelve la cantidad de caracteres sin contar el '\0'
12. Implemente un programa en el que defina un array de tres punteros a char.
a. Inicializelo en tiempo con la dirección de tres vectores que contengan los strings "cero", "uno" y "dos". Imprima las direcciones almacenadas en el vector de punteros. Imprima las leyendas utilizando el vector de punteros
b. Repita el punto anterior pero inicialice el vector de punteros en tiempo de compilación.
Indique a qué se debe que las direcciones almacenadas en el vector de punteros en ambas variantes este en zonas tan distintas.
13. Implemente un programa que imprima todos los parámetros pasados por línea de comando.
14. Implemente un programa utilizando un puntero a función ejecute una función con el siguiente prototipo que realice la suma de dos números. `int suma (int a, int b)`
15. Implemente una función que realice las cuatro operaciones básicas entre dos números de tipo float y retorne el resultado. El prototipo de la función es el siguiente
`float calculo (float opA, float opB, int op)`
Donde:
- opA y OpB son los números con los cuales se debe realizar la operación
 - op: La operación a realizar
 - 0: Realiza la suma.
 - 1: Realiza la resta.
 - 2: Realiza el producto
 - 3: Realiza la división
 - La función retorna el resultado o NAN en caso de que se ingrese una operación no válida.
- Reemplace el switch por un array de punteros a función