

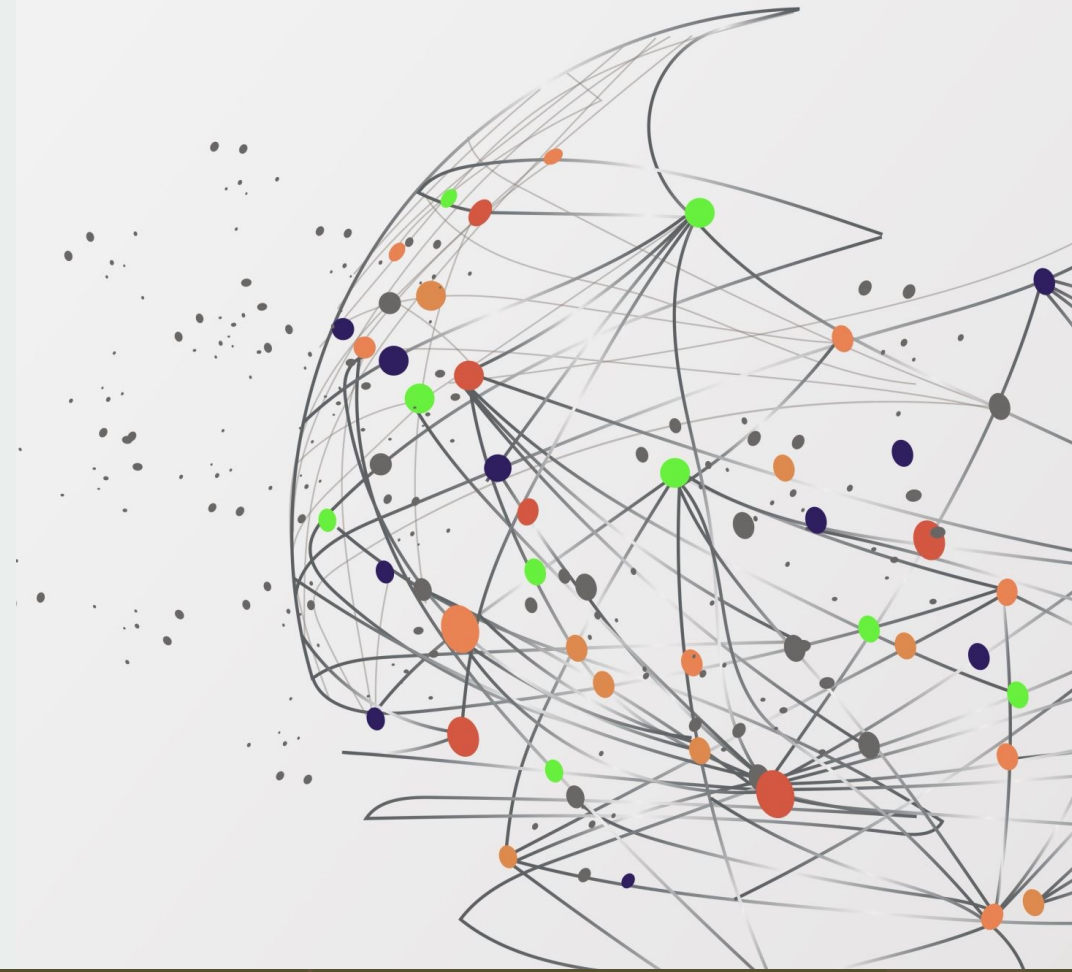


Clase Avanzados

Jose De Jesus Garfias Lopez

Introducción

- Nombre, Edad, Escuela, Lugar Nacimiento.
- Cual es tu hobby favorito?
- Cual es tu comida favorita?
- Cual es tu Anime/Serie o pelicula favorita?
- Quien es tu artista favorito?
- Cual es tu experiencia en la Olimpiada?
- Cual es el problema mas difícil que has resuelto?



Apuntadores



Que son?

- Un puntero es un tipo de variable. Almacena la dirección de un objeto en memoria y se usa para acceder a ese objeto.
- Un puntero al que no se ha asignado un valor contiene datos aleatorios.
- Se puede asignar nullptr.
- También se puede desreferenciar un puntero para recuperar el valor del objeto al que apunta.

Que son?

C++

 Copiar

```
int* p = nullptr; // declare pointer and initialize it
                  // so that it doesn't store a random address
int i = 5;
p = &i; // assign pointer to address of object
int j = *p; // dereference p to retrieve the value at its address
```

C++

 Copiar

```
MyClass* mc = new MyClass(); // allocate object on the heap
mc->print(); // access class member
delete mc; // delete object (please don't forget!)
```


Operaciones con punteros

- ***** - recuperar el valor de una referencia o puntero.
 - Retorna el tipo de variable a la que es referenciada.
- **&** - Obtener la referencia o puntero de un valor.
 - Retorna un apuntador del tipo de valor al que esta referenciando.
- El operador **sizeof** devuelve el tamaño total en bytes de un apuntador. Para determinar el número de elementos, divida el total de bytes por el tamaño de un elemento. (no se recomienda su uso)

```
int i[5]{ 1,2,3,4,5 };  
// sizeof(i) = total bytes  
int j = sizeof(i) / sizeof(i[0]);
```

Aritmética con punteros

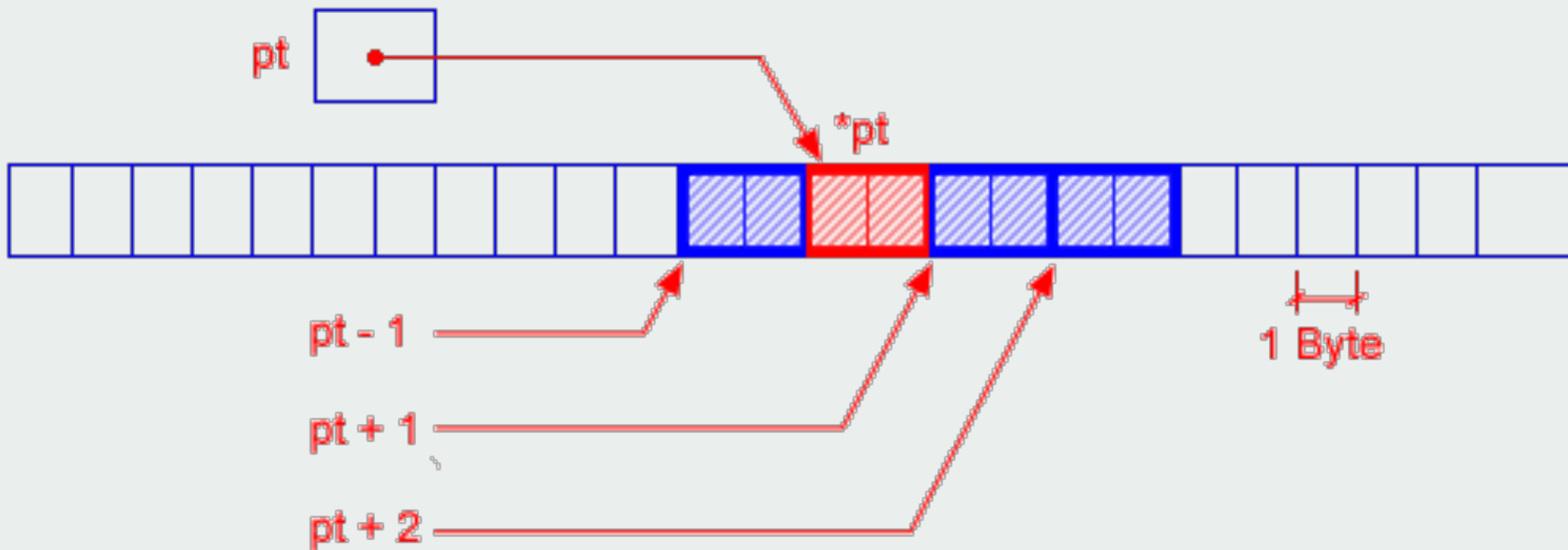
- La dirección de memoria es un valor entero.
- Se le pueden hacer todas las operaciones que un entero normal.
- No obstante se puedan realizar estas ordenes, la operación que se realiza no es totalmente transparente. Esto quiere decir que "pt++;" no suma 1 al puntero pt, ni que "pt += 5;" sume 5 a pt, ni que "pt -= 3;" o "pt--;" reste 3 ó 1 al puntero pt respectivamente.

Aritmetica con punteros

- Cuando se ejecuten esas órdenes el compilador multiplicará el valor que se desea añadir o restar al puntero por el tamaño de la variable referenciada. Esto quiere decir que, para el ejemplo anterior:
- `pt++;` El compilador lo traduce como: `pt += 1*sizeof(int);`
`pt += 5;` El compilador lo traduce como: `pt += 5*sizeof(int);`
`pt -= 3;` El compilador lo traduce como: `pt -= 3*sizeof(int);`
`pt--;` El compilador lo traduce como: `pt -= 1*sizeof(int);`

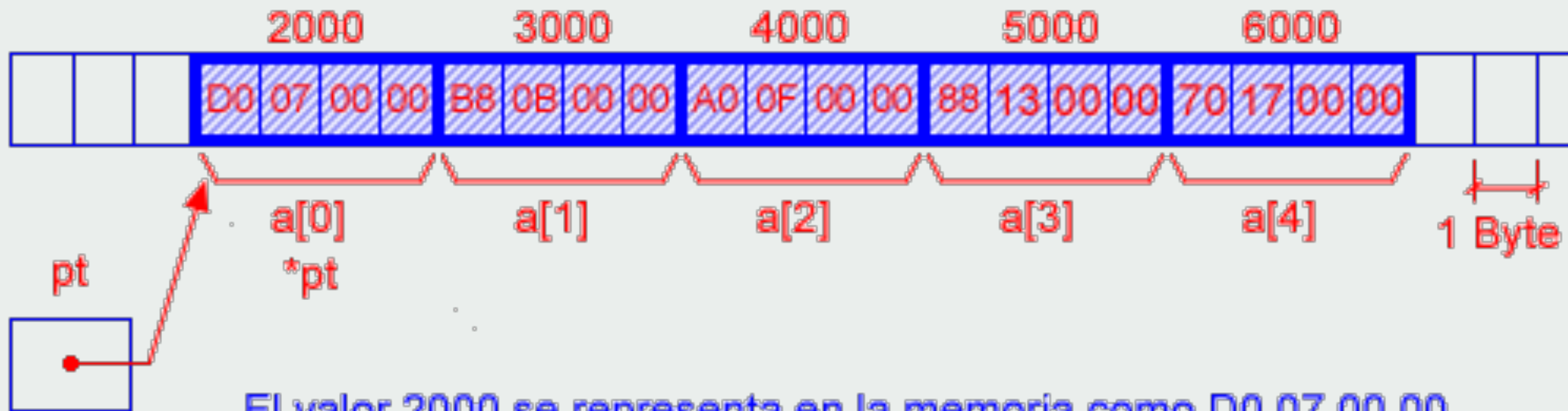
Aritmetica con punteros

- En otras palabras, sí se define "int *pt;" para versiones de C/C++ en la que un int se represente en 2 bytes (`sizeof(int) = 2`):



Aritmetica con punteros

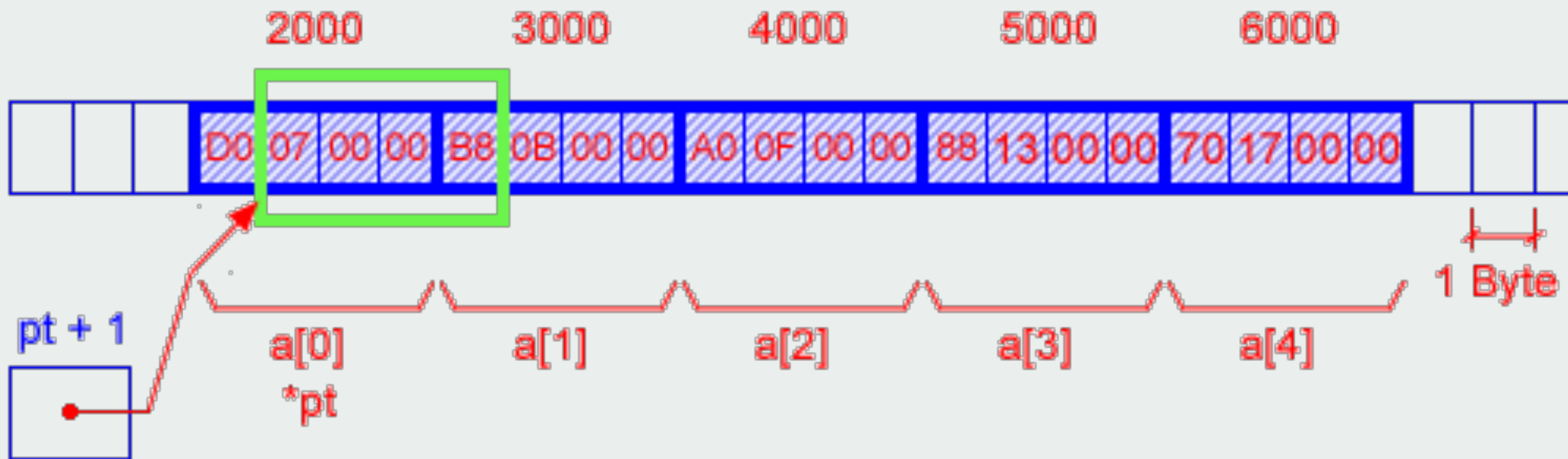
- `long a[5] = {2000, 3000, 4000, 5000, 6000};`
`long *pt;`
`pt = a;`



El valor 2000 se representa en la memoria como D0 07 00 00,
el valor 3000 como B8 0B 00 00, etc.

Aritmetica con punteros

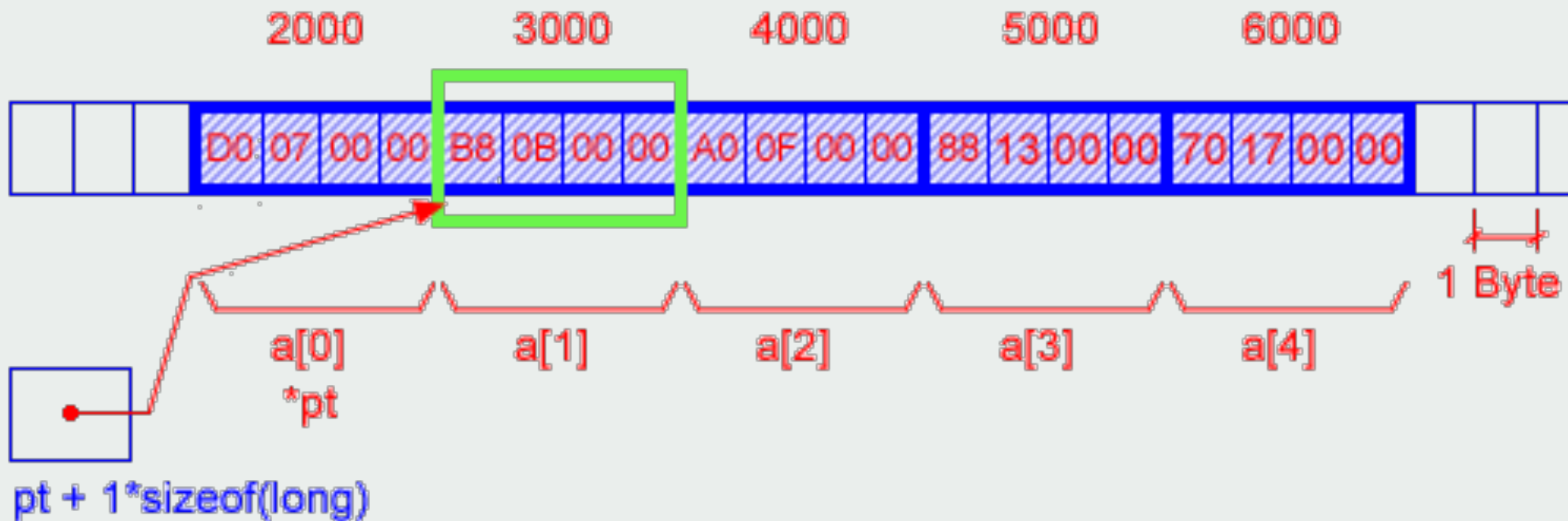
- Si cuando se realiza la operación "pt+1", se le sumase sólo 1 al puntero se tendría:



La representación en memoria de 07 00 00 B8 es equivalente al número -1207959545, un valor incoherente.

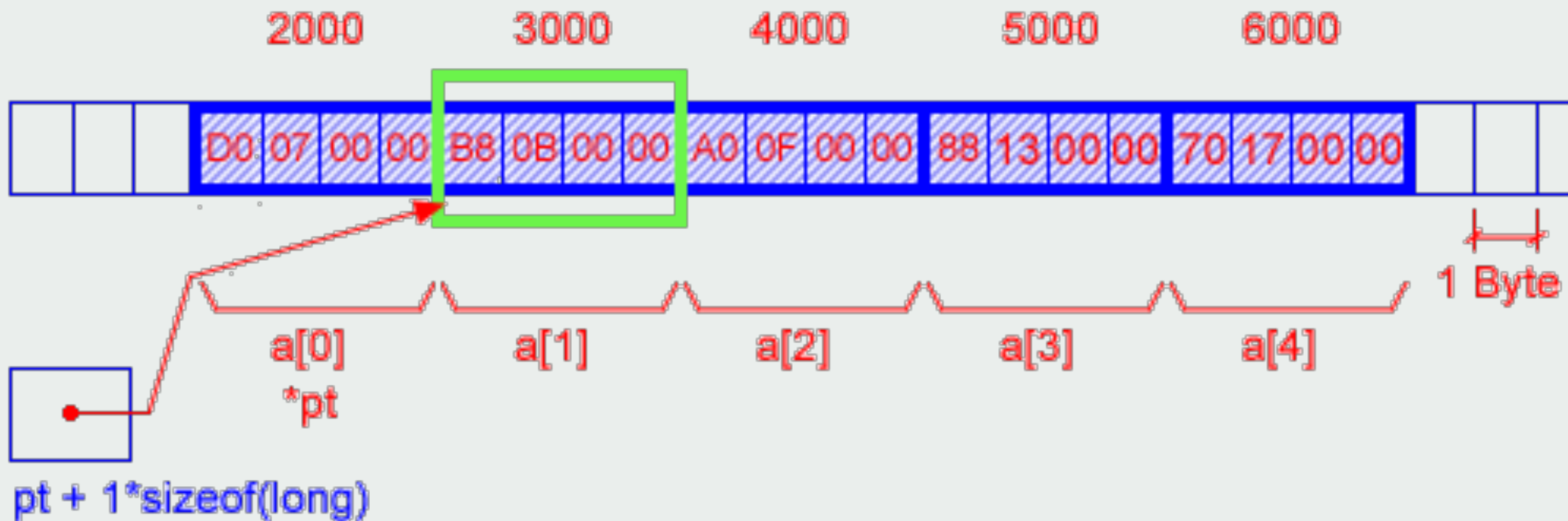
Aritmetica con punteros

- Pero si se le sumase $1 * \text{sizeof}(\text{long})$ al puntero se tendría:



Aritmetica con punteros

- Pero si se le sumase $1 * \text{sizeof}(\text{long})$ al puntero se tendría:



Paso de punteros en funciones

- Una función se le puede pasar como parámetro la dirección de memoria de una variable, esta dirección puede ser tomada por un puntero quien a su vez podrá modificar la variable referenciada, esta última operación modificaría el contenido de la variable cuya dirección se pasó como parámetro.

Ejercicio

Función C++ que reciba la dirección de inicio de una cadena de caracteres y devuelva el número de vocales que hay en la cadena.

- `int cuenta_vocales(char *s);`