

Unidad 6: Punteros parte 2

Laboratorio – Sesión 1 y/o 2

<http://bit.ly/2p3fgiD>

Profesores:

< Juan Flores Moroco >

Telegram:

1. Configurar tu cuenta

2. Link: <http://bit.ly/2OW5Ss9>

SISAP:



SISAP 2018
October 7-9
Lima, Perú

Evento: SISAP 2018 – 11th International Conference on Similarity Search and Applications

Fechas: October 7-9 Lima, Perú

Resumen: <http://www.sisap.org/2018/>

The 11th International Conference on Similarity Search and Applications (SISAP) is an annual forum for researchers and application developers in the area of similarity data management. It aims at the technological problems shared by numerous application domains, such as data mining, information retrieval, multimedia, computer vision, pattern recognition, computational biology, geography, biometrics, machine learning, and many others that make use of similarity search as a necessary supporting service.

Inscripciones: https://eventos.spc.org.pe/spire2018/registration_sisap.html

SPIRE:

SPIRE 2018: 25th International Symposium on String Processing and Information Retrieval

Fechas: October 9-11 Lima, Perú

Resumen: <https://eventos.spc.org.pe/spire2018/venue.html>

SPIRE 2018 is the 25th edition of the annual Symposium on String Processing and Information Retrieval. SPIRE has its origins in the South American Workshop on String Processing, which was first held in Belo Horizonte, Brazil, in 1993. Since 1998 the focus of the workshop has also included information retrieval, due to its increasing relevance to and inter-relationship with string processing.

SPIRE 2018 will be held in UTEC Lima, Peru.

Inscripciones: <https://eventos.spc.org.pe/spire2018/registration.html>

Logro de la sesión:

Al finalizar la sesión, los alumnos desarrollan sus programas utilizando punteros y arrays dinámicos.

Punteros



Uso de la memoria primaria en C++

Segmento de Información Externa

Argumento externos (argc, argv)

Pila (Stack)

Memoria automática donde se asigna las variables estáticamente.



Montón (Heap)

Memoria donde se asigna variables dinámicamente.

Segmento Estático y Global

Memoria donde se asigna variables globales y tipo static.

Segmento de Código

Memoria donde se guarda el segmento de código

Matrices dinámicas

Se crea una matriz de 4 x 7 en el heap:

	0	1	2	3	4	5	6
0	23	34	45	56	12	50	6
1	76	43	98	35	8	63	12
2	67	2	32	73	12	15	18
3	38	5	3	79	16	20	24

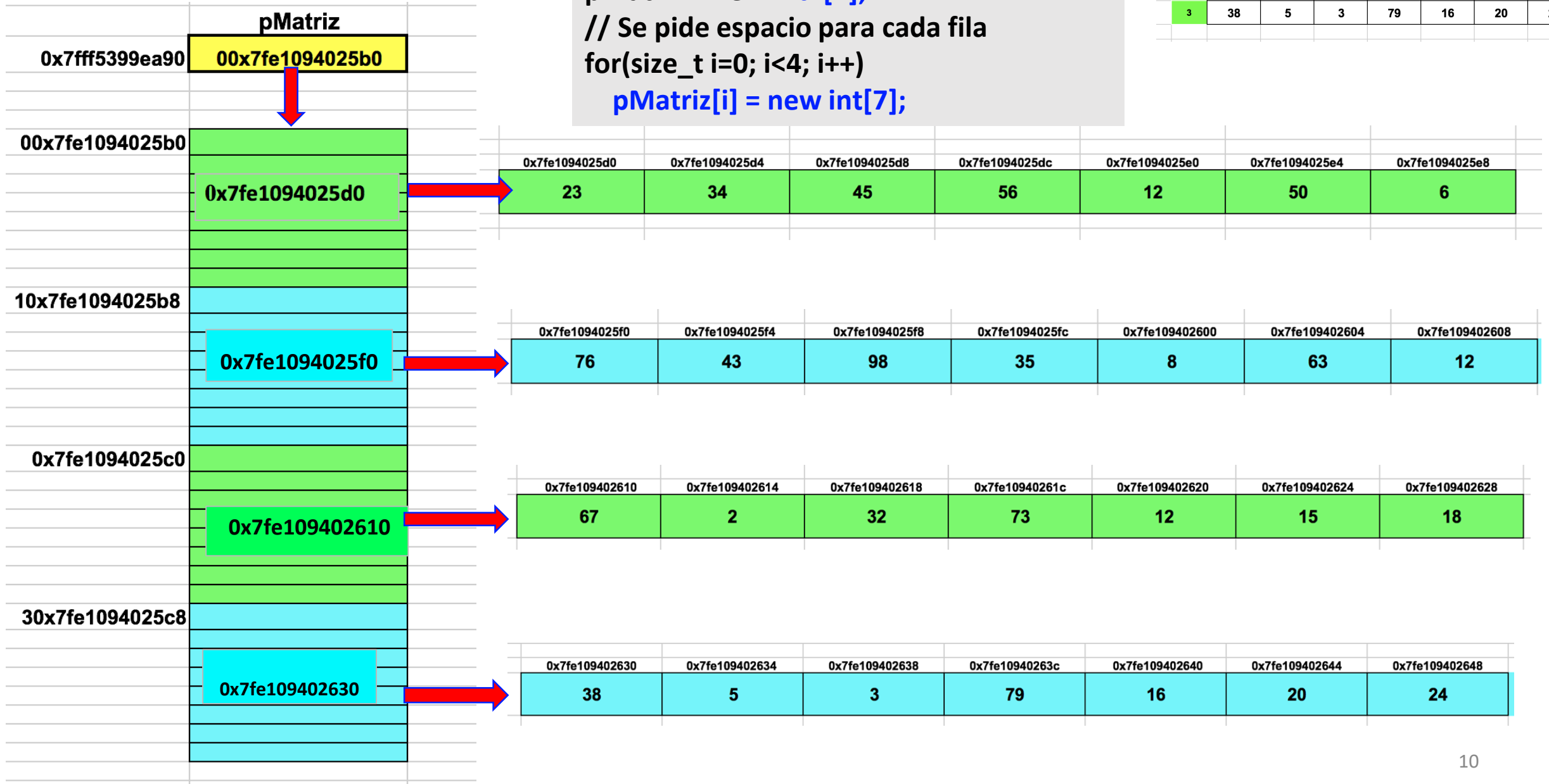
4x7

```

int **pMatriz= nullptr;
pMatriz = new int*[4];
// Se pide espacio para cada fila
for(size_t i=0; i<4; i++)
    pMatriz[i] = new int[7];

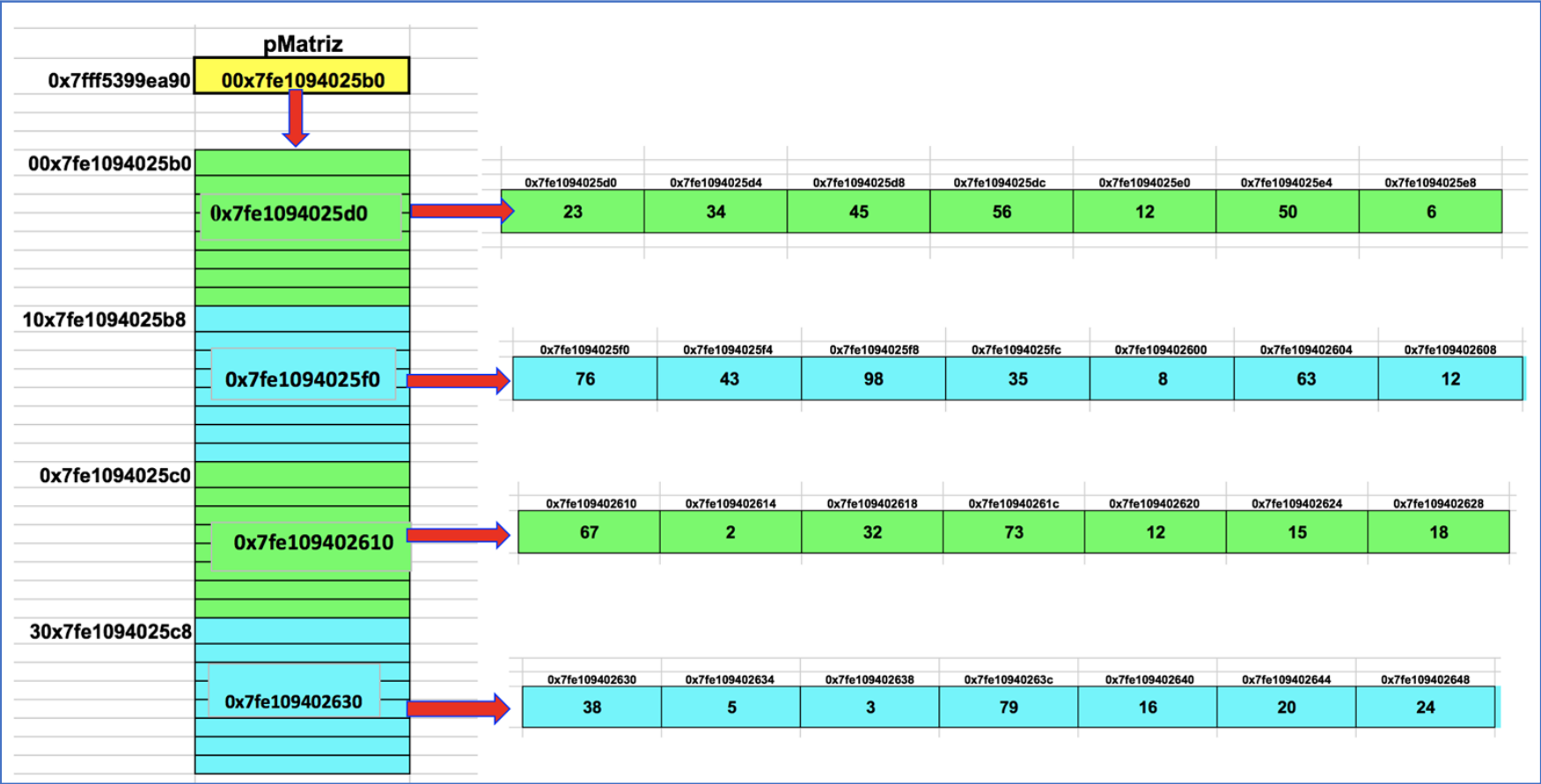
```

	0	1	2	3	4	5	6
0	23	34	45	56	12	50	6
1	76	43	98	35	8	63	12
2	67	2	32	73	12	15	18
3	38	5	3	79	16	20	24



En la matriz definida dinámicamente, y suponiendo que el compilador del Clion utiliza 4 bytes para almacenar un dato int y 8 bytes para almacenar un puntero.

¿ Cuántos bytes en total ocupa la matriz?

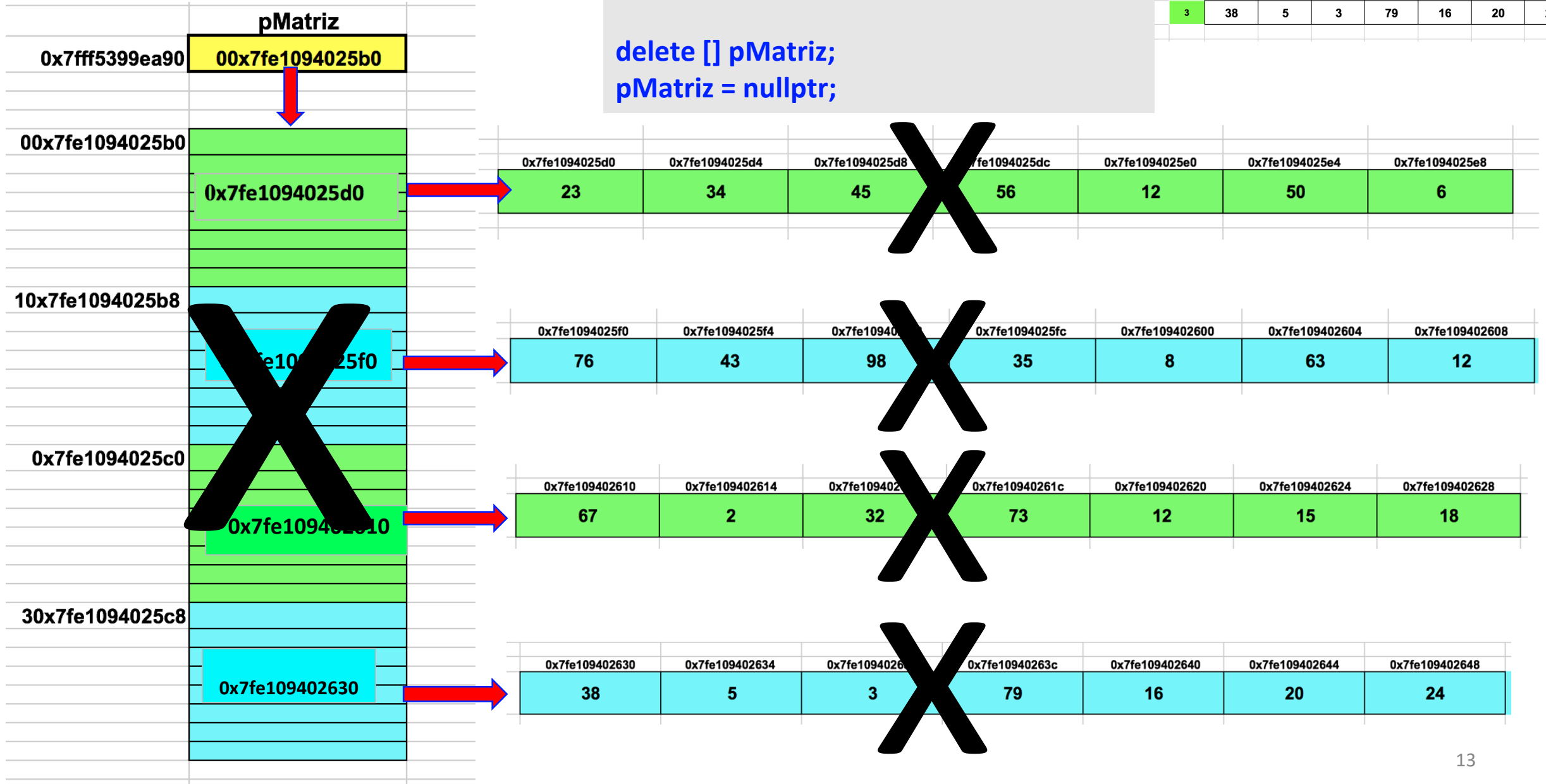


Para liberar memoria

```
for(size_t i=0; i<4; i++)
    delete [] pMatriz[i]
```

```
delete [] pMatriz;
pMatriz = nullptr;
```

	0	1	2	3	4	5	6
0	23	34	45	56	12	50	6
1	76	43	98	35	8	63	12
2	67	2	32	73	12	15	18
3	38	5	3	79	16	20	24



Ejemplo 1:

Desarrolle un programa que permita multiplicar dos matrices de números enteros.

El programa debe solicitar al usuario que se ingrese el orden de la matriz 1 y de la matriz 2. Debe verificar si matemáticamente es posible realizar la multiplicación de matrices.

Si no es posible el programa termina.

Si se puede realizar la multiplicación de matrices, entonces el programa debe realizar lo siguiente:

- **Pedir espacio en el heap para la primera matriz.**
- **Pedir espacio en el heap para la segunda matriz.**
- **Generar datos de manera aleatoria para cada matriz.**
- **Imprimir ambas matrices.**
- **Realizar la multiplicación de matrices, para ello previamente se habrá tenido que dimensionar la matriz resultado.**
- **Finalmente imprimir la matriz resultado.**

Realice el programa utilizando funciones y distribuyendo en código en archivos *.cpp y *.h