

Universidad Tecnológica Centroamericana  
UNITEC

19 de julio del 2019

**Sesión #1**

Laboratorio de Programación III

Javier Cano

# Desarrollo de la práctica

Para el desarrollo satisfactorio del laboratorio, siga al pie de la letra cada instrucción que a continuación se le presenta.

# Objetivos del laboratorio

* Desarrollar la lógica mediante la resolución de problemas.
* Utilizar los flujos estándares de impresión y obtención de datos.
* Manejo y uso de todas y cada una de las estructuras de control.
* Modelar soluciones a problemas mediante funciones y apuntadores en C++.
* Hacer uso de matrices en C++.

# Ejercicio 1

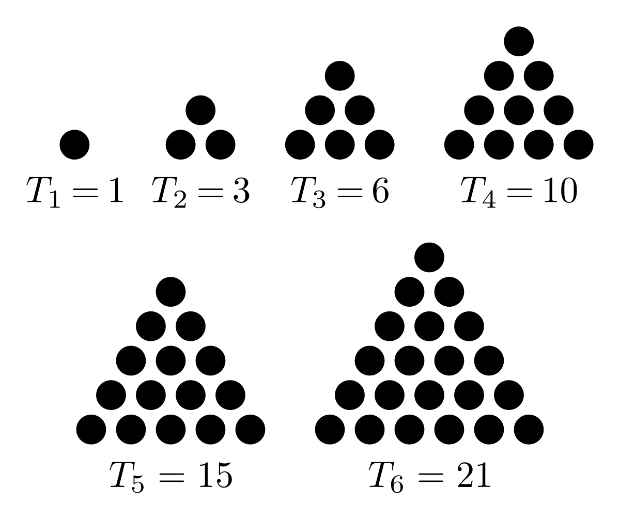
Un **punto de silla** es un punto (en este caso de una matriz) donde su magnitud es mínima en la fila y máxima en la columna.

Dada una matriz M con dimensiones 5x5, crear una función que reciba la matriz M y que realice las siguientes tareas:

1. Mostrar la matriz M que se envió a la función
2. Mostrar todos los puntos de silla que puede haber en la matriz M, con la magnitud y las coordenadas (x, y) donde se encontró el punto de silla
3. Mostrar la cantidad de puntos de silla encontrados en la matriz M

# Ejercicio 2

Los **números triangulares** son una secuencia de números que siguen un patrón, donde si tomamos los números y los agrupamos en un triángulo, estos pueden formar un triángulo equilátero:



Teniendo esto en cuenta, su tarea consta de crear una función que identifique si un numero N es o no es triangular. Si N es triangular, comunicarlo por medio de un mensaje. Si N no es triangular, siempre comunicarlo por medio de un mensaje y, además, mostrar el numero triangular menor a N más cercano. Por ejemplo:

* Si el usuario ingresa 6, comunicar que es un numero triangular.
* Si el usuario ingresa 23, comunicar que no es triangular y que 21 es el numero triangular más cercano.

# Ejercicio 3

Dado un arreglo cualquiera de caracteres **sin repeticiones**, desarrollar una función que muestre en pantalla todas las secuencias posibles de estos caracteres.

Por ejemplo:

**AABC** no es válido.

**ABC** produce:

* **ABC**
* **ACB**
* **BAC**
* **BCA**
* **CAB**
* **CBA**

# Ponderación

|  |  |
| --- | --- |
| Elemento | Puntaje |
| Ejercicio 1 | 2 |
| Ejercicio 2 | 3 |
| Ejercicio 3 | 3 |
| Utilizar una función recursiva | 1 |
| Mínimo de 10 commits en GitHub | 1 |

# Comentarios adicionales

Subir el documento como un archivo de **Word (.docx),** incluyendo su nombre completo, número de cuenta y repositorio de GitHub.

**Asegúrense de probar su programa y de que el código sea legible.**

Se aplicarán penalizaciones por quiebres en el código, interrupciones o errores inesperados en el programa y por mala indentación del código. Por cada una de las penalizaciones encontradas se les restara un punto de la nota final de su laboratorio.

Si el código no es legible para otras personas o está mal indentado, entonces los instructores no nos molestaremos por tratar de entender que hace su programa en caso de que se presente un error en el mismo.