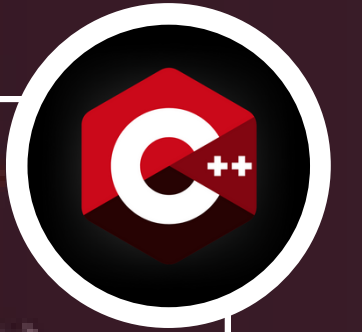


BIENVENIDOS



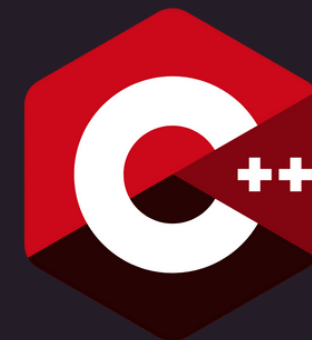
MENTORÍA

PROGRAMACIÓN II

C++

SESIÓN 3





SESIÓN 3

**Punteros,
memoria dinámica
y PC1**





Flujo de la clase



Resumen de la sesión

1

Memoria
Dinámica

2

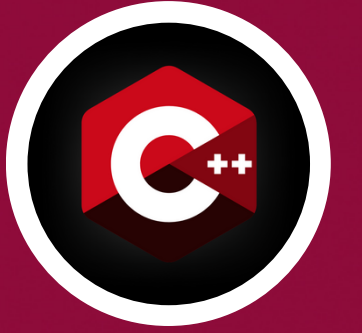
Punteros

3

Arrays estáticos

4

Excercises



MEMORIA DINÁMICA

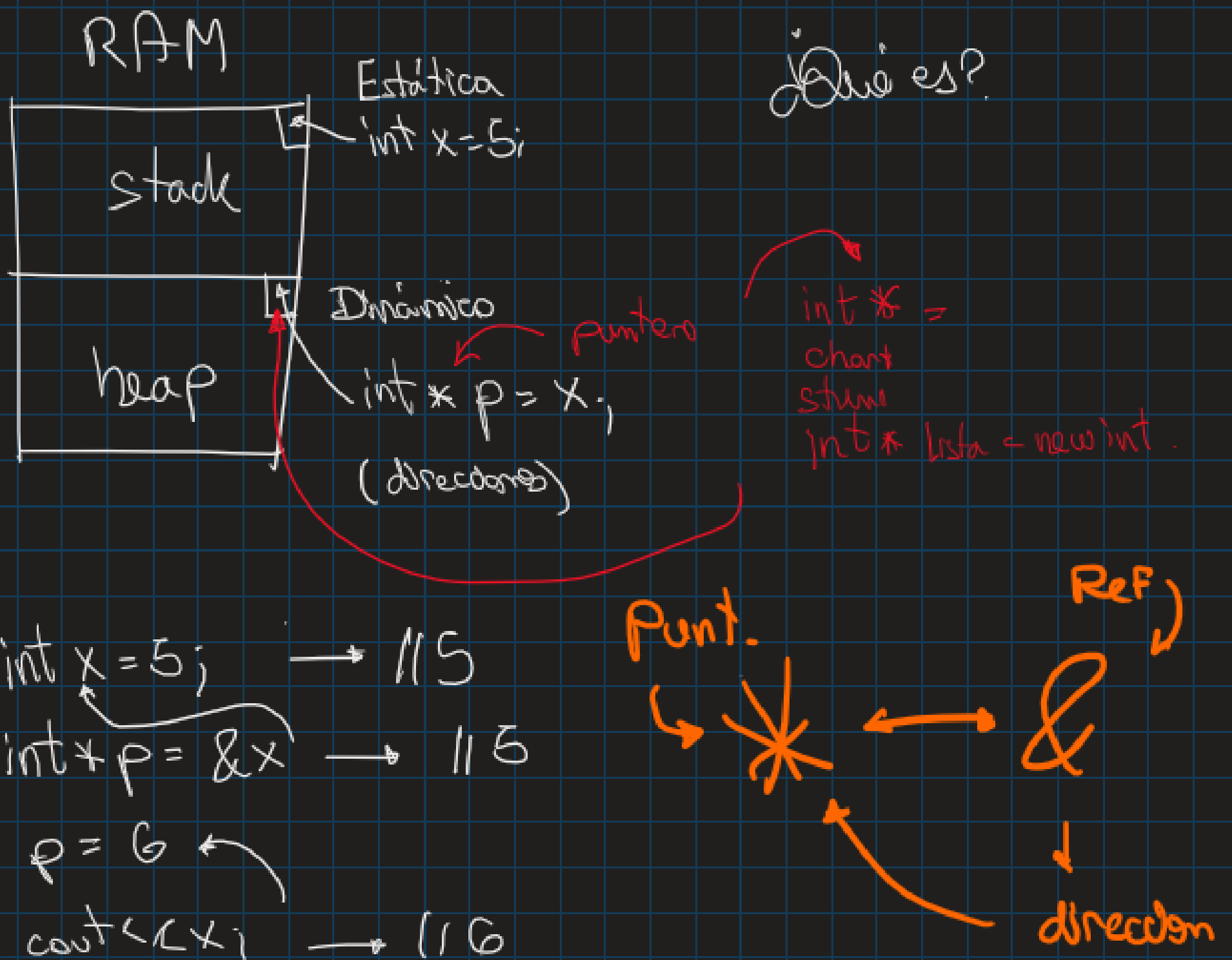
Segmento Estático y Global

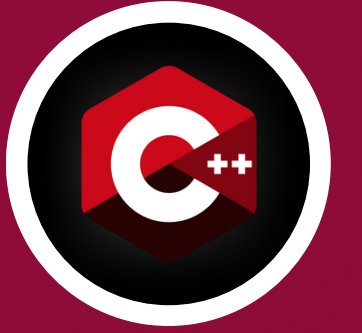
Segmento de Información Externa

Segmento de Código

Pila (Stack)

Montón (Heap)





PUNTEROS

PUNTEROS

Un puntero es un tipo de dato

Un puntero de un tipo apunta a un objeto/variable del mismo tipo

Un puntero almacena direcciones de objetos

Pueden existir ciclos de punteros

```
int x = 5;
cout << x << endl;
cout << &x << endl;

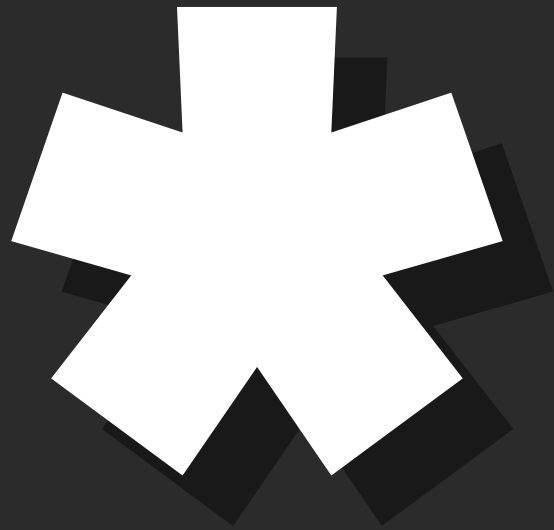
int* p = &x;

cout << p << endl;
cout << *p << endl;

*p = 6;

cout << x << endl;
```


PUNTEROS



- Un puntero es un objeto, una referencia no
- Un puntero apunta a una referencia de un objeto
- Ambos hacen modificaciones al objeto inicial
- Los punteros se almacenan en el heap, las referencias en el stack

DEREFERENCIA

DIRECCIONES



REFERENCIA



ARRAYS ESTÁTICOS

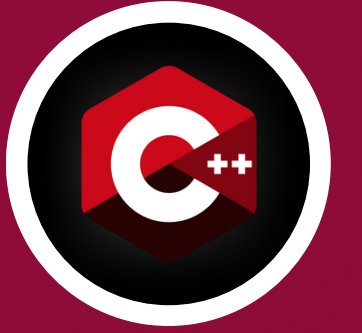
ARRAYS ESTÁTICOS

Es una colección de datos de un mismo tipo

Los valores se almacenan en el Stack

Se debe definir su tamaño en tiempo de compilación

```
int arr1[3];  
int arr2[3] = { [0]: 1, [1]: 2, [2]: 3};  
int arr3[5]{ [0]: 1, [1]: 2, [2]: 3};  
  
int size = 4;  
char arr4[size]['a', 'b', 'c'];
```

PRACTISE PC1

Se le pide:

Elaborar una función en lenguaje C que calcule las coordenadas del cliente de menor riesgo y las coordenadas del cliente de mayor riesgo.

Esta función deberá recibir como parámetros al arreglo de horas, al arreglo de velocidad, y deberá calcular las coordenadas (x_1, y_1) del cliente de menor riesgo y las coordenadas (x_2, y_2) del cliente de mayor riesgo de tal forma que estas dos coordenadas puedan ser comunicadas hacia afuera de la función.

Desde la main, usted deberá llamar a dicha función, y enviar **desde adentro de la función**, la información correspondiente a las coordenadas (x_1, y_1) del cliente de menor riesgo y las coordenadas (x_2, y_2) del cliente de mayor riesgo **hacia main**.

Una vez que reciba las coordenadas de estos puntos en su función main, deberá imprimirlos.

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

3. (7 points) Rodar Números

Utilizando punteros, implemente una función que modifique el orden de una secuencia de números. Para ello considere lo siguiente:

- Crear un array estático con los número del 1 al 6.
- Crear una función para imprimir los elementos del array en una linea y separados por un espacio.
- Crear una función que modifique la posición de los elemento del array original según el número entero ingresado.
 - Si el valor es positivo, trasladar los elementos hacia la izquierda.
 - Si el valor es negativo, trasladar los elementos hacia la derecha.

Algunos ejemplos de diálogo de este programa serían:

Listing 5: Ejemplo 1


```
Array original: 1 2 3 4 5 6
Ingresa un numero: 3
Array modificado: 4 5 6 1 2 3
```

cout<<"Thank
you"

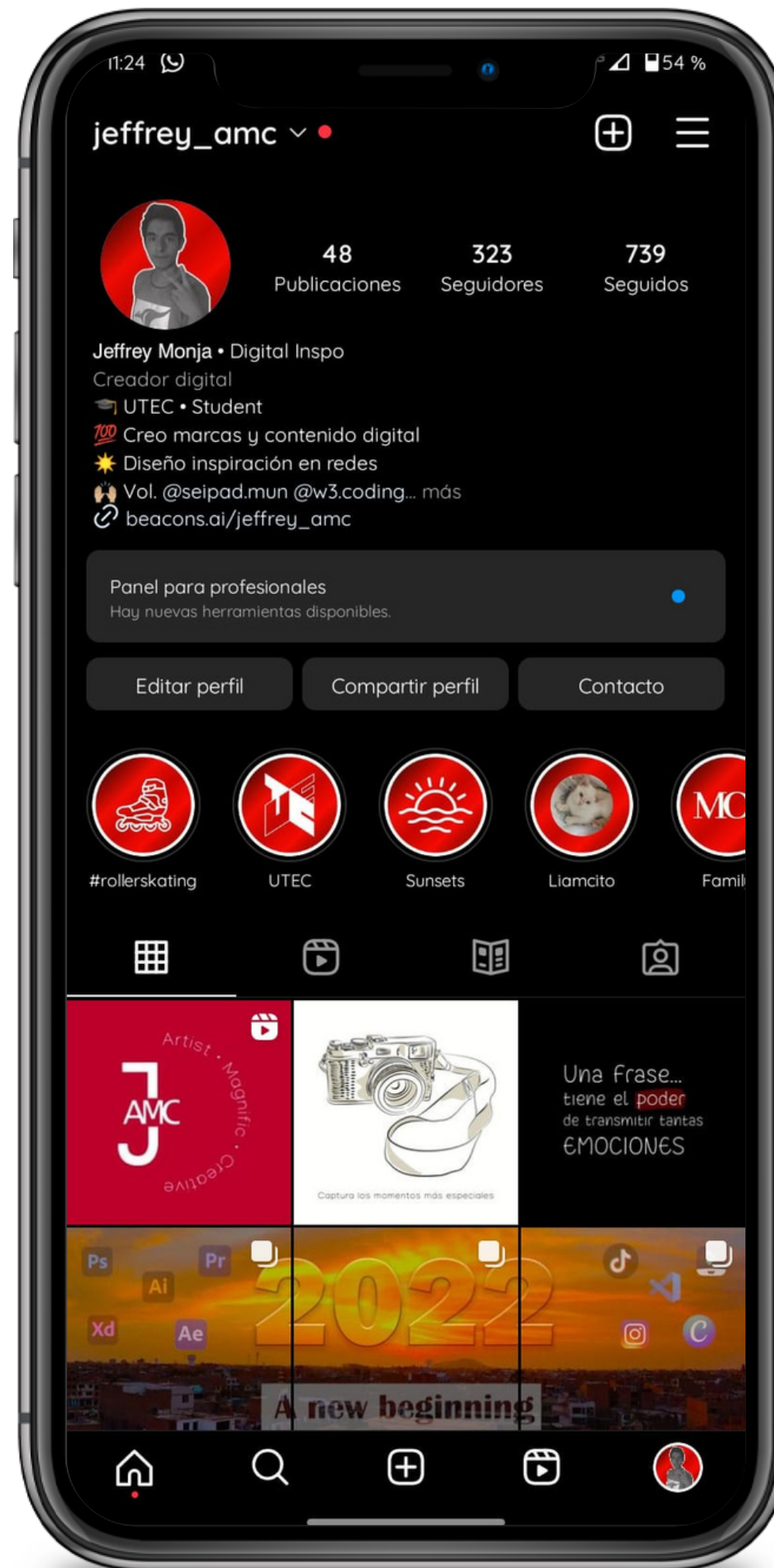
 Jeffrey Monja
Castro

 @jeffrey_amc

 Jeffrey Monja
Castro

 942482139

 jeffrey.monja
@utec.edu.pe



ありがとう!

 @jeffrey_amc