

CS1112: Programación II

Unidad 3: Punteros

Sesión de Laboratorio - 5A

Profesores:

María Hilda Bermejo mbermejo@utec.edu.pe

Estanislao Contreras econtreras@utec.edu.pe

Jorge Villavicencio jvillavicencio@utec.edu.pe

Edson Mendiola emendiola@utec.edu.pe

Ian Paul Brossard ibrossard@utec.edu.pe

Jose Chavez jchaveza@utec.edu.pe

Julio Yarasca jyarascam@utec.edu.pe

Percy Quevedo pquevedo@utec.edu.pe

Wilder Nina wnina@utec.edu.pe

José Fiestas jfiestas@utec.edu.pe

Material elaborado por:

Maria Hilda Bermejo, Ruben Rivas



Índice:

- **Unidad 3: Punteros**
 - Definicion
 - Manejo de memoria dinámica

3.2

Unidad 3: Memoria dinámica (arreglos)

UTEC

Logro de la sesión:

Al finalizar la sesión, los alumnos podrán:

- Desarrollar programas utilizando punteros.
- Implementará programas en C++, utilizando punteros y creando arrays unidimensionales de manera dinámica.

Arreglos dinámicos

Ejemplo 1:

Escriba un programa que permita leer como dato un número entero mayor a 15, el cual representará el número de alumnos de un salón de clase.

El programa deberá llenar un array con N números aleatorios, para luego:

1. Imprimir solo las notas que están por encima del promedio
2. Imprimir la mayor nota
3. Imprimir el promedio eliminando la menor nota. Si el valor de la nota menor se repite se deberá eliminar todas las ocurrencias y realizar el cálculo del promedio con las notas restantes.

Utilice funciones y punteros dinámicos

Ejemplo 2:

Realice un programa que permita leer como dato un número que representa la cantidad de elementos que tendrá un array dinámico. Luego realice lo siguiente:

- Crear el array, llenarlo con números aleatorios entre 0 y 999.
- Imprimir el array
- Generar a partir de ese array dos nuevos array dinámicos, el primero con los múltiplos de 5 y el segundo con los múltiplos de 7 que tenga el primer array.

Ejemplo 3:

Desarrollar un programa que permita generar aleatoriamente un array de números enteros, el tamaño del array es un dato que ingresará el usuario.

Luego el programa imprimirá el array y pedirá al usuario que ingrese el número que desea buscar en el array.

El programa luego tendrá que indicar cuántas veces está el dato a buscar en el array y además creará un segundo array con las posiciones en que el dato se encuentra en el array.

Si el dato no está en el array, entonces se imprimirá solo el mensaje que indica que el dato no se encuentra en el array.

Ejemplo 4:

Desarrollar un programa que permita generar aleatoriamente un array de números enteros del 1 al 30, el tamaño del array es un dato que ingresará el usuario.

Calcular los valores mostrados en el cuadro, donde:

- N = Cantidad Total de valores.
 - Xi = Valor X almacenado en la posición i.
- NOTA: “i” no es equivalente el subíndice del array.

Los demás valores son deducibles de estos valores básicos.

Nombre	Fórmula
Media	$\bar{x} = \frac{1}{N} \cdot \sum_{i=1}^N x_i$
Moda	Datos x_i más repetidos.
Mediana	$\frac{x_{\frac{N+1}{2}}}{2} \text{ si } N \text{ impar}$ $\frac{1}{2} \cdot \left(x_{\frac{N}{2}} + x_{\frac{N}{2}+1} \right) \text{ si } N \text{ par}$
Desviación respecto de la media	$D_i = x_i - \bar{x} $
Desviación media	$D_m = \frac{1}{N} \cdot \sum_{i=1}^N x_i - \bar{x} $
Varianza	$\sigma^2 = \frac{1}{N} \cdot \sum_{i=1}^N (x_i - \bar{x})^2$
Desviación estándar	$\sigma = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N (x_i - \bar{x})^2}$

Resumen

En esta sesión aprendiste a:

1. Acceder al **stack** y **heap** de la memoria por medio de los punteros.
2. Usar **arreglos** que son tipos de datos compuestos que permiten almacenar una **colección de datos del mismo tipo** en forma **secuencial y consecutiva**.
3. Comprender que las variables almacenadas en la memoria automática o **stack** se reservan y liberan de **forma automática**.
4. **Reservar** y **liberar** de **forma manual** en la memoria dinámica o **heap** por medio de los operadores **new** y **delete**.

Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall

¡Nos vemos en la siguiente
clase!

