



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Alumno:

Ramírez Monjaraz Miguel Ángel

Profesor:

[Pedro Nunez Yepiz](#)

Ingeniería en computación

Grupo: 432

Materia:

Programación - Estructurada

Tema:

Actividad 8

Arreglos - Funciones

Fecha de entrega:

03 de Octubre del 2023



Ingeniero en computación

Materia: Programación Estructurada / Clave 36276

Alumno: Miguel Angel Ramirez Monjaraz

Matrícula: 372205

Maestro: Pedro Núñez Yépiz

Actividad No. : 8 - SEMANA 8

Tema - Unidad : Arreglos

Unidad 1

Ensenada Baja California a 03 de Octubre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

Los arrays son variables estructuradas, donde cada elemento se almacena de forma consecutiva en memoria. Los vectores son un tipo de array (arreglos), estos se clasifican según su dimensión, es decir van desde unidimensional hasta arreglos denominados matrices

2. COMPETENCIA

Que el alumno reconozca el funcionamiento básico de un vector, su manipulamiento, y cada componente que lo distingue de una cadena de caracteres, además en esta práctica lo complementará con la teoría aprendida en las prácticas pasadas.

3. FUNDAMENTOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Los vectores son un tipo de array (arreglos). Son, de hecho, un array de una sola dimensión y forman parte de la amplia variedad de estructuras de datos que nos ofrece C++, siendo además una de las principales y más útiles estructuras que podremos tener como herramienta de programación. Los vectores o arrays o arreglos de una dimensión (como los quieras llamar), son utilizados para almacenar múltiples valores en una única variable. En un aspecto más profundo, este tipo de arrays (vectores), permiten almacenar muchos valores en posiciones de memoria continua, lo cual permite acceder a un valor u otro de manera rápida y sencilla. Estos valores pueden ser números, letras o cualquier tipo de variable que deseemos incluso tipos de datos complejos.

Para declarar un vector, necesitaremos saber el tipo de los datos que irán al interior de este, es decir, serán número enteros, o número decimales, etc. necesitamos también, como siempre, un nombre para el vector y un tamaño máximo. La sintaxis para declarar un vector en C++ es la siguiente:

```
tipo_de_dato nombre_del_vector[tamaño];
```

Seguiremos trabajando con las funciones, y recordando que son, estas no son más que una herramienta indispensable para el programador, tanto las funciones creadas por él mismo como las que le son proporcionadas por otras librerías, cualquiera que sea el caso, las funciones permiten automatizar tareas repetitivas, encapsular el código que utilizamos, e incluso mejorar la seguridad, confiabilidad y estabilidad de nuestros programas. Dominar el uso de funciones es de gran importancia, permiten modularizar nuestro código, separarlo según las tareas que requerimos, por ejemplo una función para abrir, otra para cerrar, otra para actualizar, etc. básicamente una función en nuestro código debe contener la implementación de una utilidad de nuestra aplicación, es decir que por cada utilidad básica (abrir, cerrar, cargar, mover, etc.) sería adecuado tener al menos una función asociada a ésta.

Recordemos que una función siempre retorna algo, por lo tanto es obligatorio declarar un tipo (el primer componente de la sintaxis anterior), luego debemos darle un nombre a dicha función, para poder identificarla y llamarla durante la ejecución, después al interior de paréntesis, podemos poner los argumentos o parámetros. Luego de la definición de la "firma" de la función, se define su funcionamiento entre llaves; todo lo que esté dentro de las llaves es parte del cuerpo de la función y éste se ejecuta hasta llegar a la instrucción return.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Hay algunos detalles respecto a los argumentos de una función, veamos:

1. Una función o procedimiento pueden tener una cantidad cualquier de parámetros, es decir pueden tener cero, uno, tres, diez, cien o más parámetros. Aunque habitualmente no suelen tener más de 4 o 5.
2. Si una función tiene más de un parámetro cada uno de ellos debe ir separado por una coma.
3. Los argumentos de una función también tienen un tipo y un nombre que los identifica. El tipo del argumento puede ser cualquiera y no tiene relación con el tipo de la función.

4. PROCEDIMIENTO

MENÚ

- 1.- LLENAR VECTOR 1 (MANUALMENTE)
- 2.- LLENAR VECTOR 2 ALEATORIAMENTE
- 3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)
- 4.- IMPRIMIR VECTORES
- 5.- LLENA MATRIZ 4 X 4
- 6.- IMPRIMIR MATRIZ
- 0.- SALIR

NOTA: EL PROGRAMA DEBERÁ REPETIRSE CUANTAS VECES LO DESEE EL USUARIO

NOTA 2: EL VECTOR 1 DE 10 POSICIONES, NÚMEROS DEL 30 AL 70

NOTA 3: EL VECTOR 2 DE 10 POSICIONES CON NÚMEROS GENERADOS ALEATORIAMENTE DEL 1 AL 20 (SIN REPETIR)

NOTA 4: EL VECTOR 3 DE 20 POSICIONES, CON LOS DATOS DEL ARREGLO 1 Y ARREGLO 2

NOTA 5: MATRIZ 4 X 4 LLENARLA CON LOS DATOS DEL VECTOR1 Y VECTOR 2.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES

Para esta práctica se logró su objetivo que era comprender el funcionamiento de un vector, esto debido a que fue entendible gracias a los demás temas vistos con anterioridad, que van desde estructuras de control, ciclos, funciones, etc. Podemos concluir que el uso de arrays en el lenguaje de programación C es fundamental y poderoso, debido a que proporcionan una forma eficiente de almacenar y acceder a datos del mismo tipo en una estructura de datos unidimensional o multidimensional.

6. ANEXOS

<https://docs.google.com/document/d/1EPSUdqX-QVsZ34ecTg6pu-Bsofuxr7wGbNgEBScpPDM/edit>



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación.Sznajdleder, P. A. (2017)..

Buenos Aires,Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN:9688804711

Programación en C.Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España:McGraw-Hill.

ISBN: 8448130138