

Universidad Mesoamericana

Facultad de Ingeniería

Estructuras Lógicas I

Ing. Romeo Roberto De Paz Aquí



TABLAS DE VERDAD EN C++

Adrián Pablo Jose Girón Franco 20230836

Diego José Alegría Orós 202308033

Josué Fernando Vázquez yax 202308030

Quetzaltenango, 24 de mayo 2023



INTRODUCCION

Una tabla de verdad es una herramienta fundamental en la lógica y la teoría de circuitos digitales. Representa de manera sistemática todas las posibles combinaciones de valores de entrada y sus correspondientes resultados de salida en una función lógica. Consta de columnas que muestran los valores de verdad de las variables de entrada y una columna adicional que indica el resultado lógico de la función. Las tablas de verdad permiten analizar y comprender el comportamiento de las funciones lógicas y son utilizadas en el diseño y evaluación de circuitos digitales, y lo podrá visualizar en un programa hecho en c++

ESTRUCTURA Y FUNCIONAMIENTO D;EL PROGRAMA

1. Se incluyen las bibliotecas `iostream` y `conio.h`, que son necesarias para realizar la entrada y salida de datos en la consola.
2. Se declara la función `main()` como punto de entrada del programa.
3. Se definen las variables `op1` y `op2` como enteros, que se utilizarán para almacenar las opciones de operadores lógicos ingresadas por el usuario.
4. Se crean los arreglos `a`, `b` y `c` de tipo booleano, que representan los posibles valores de verdad para las variables `A`, `B` y `C`.
5. Se muestra un mensaje en la consola solicitando al usuario que ingrese la opción del primer operador lógico.
6. Se lee la opción ingresada por el usuario y se almacena en la variable `op1`.
7. Se muestra un mensaje en la consola solicitando al usuario que ingrese la opción del segundo operador lógico.
8. Se lee la opción ingresada por el usuario y se almacena en la variable `op2`.
9. Se muestra en la consola la palabra "Tabla" para indicar el comienzo de la tabla de verdad.
10. Se utiliza una estructura de control `switch` para mostrar el nombre del primer operador lógico seleccionado por el usuario en función del valor de `op1`.
11. Se utiliza otra estructura de control `switch` para mostrar el nombre del segundo operador lógico seleccionado por el usuario en función del valor de `op2`.
12. Se muestra en la consola la cabecera de la tabla, que indica las variables `A`, `B` y `C`.
13. Se utiliza un bucle `for` anidado para generar todas las combinaciones posibles de valores de verdad para las variables `A`, `B` y `C`.
14. Dentro del bucle, se muestra en la consola la combinación actual de valores de verdad para `A` y `B`, junto con los operadores lógicos seleccionados por el usuario.
15. Se utiliza una estructura de control `switch` para calcular el resultado de la operación lógica correspondiente a `op1` utilizando los valores actuales de `A` y `B`. El resultado se almacena en la variable `result`.
16. Se utiliza otra estructura de control `switch` para calcular el resultado final de la operación lógica correspondiente a `op2` utilizando el resultado anterior `result` y el valor actual de `C`. El resultado final se almacena nuevamente en la variable `result`.
17. Se muestra en la consola el resultado final de la operación lógica.
18. El bucle `for` continúa generando todas las combinaciones posibles de valores de verdad para `A`, `B` y `C`, y repite los pasos del 14 al 17.
19. Después de que se generan todas las combinaciones posibles, el programa espera a que el usuario presione cualquier tecla para terminar la ejecución.
20. Finalmente, el programa retorna 0 para indicar que se ha ejecutado correctamente.

Link del Proyecto

<https://github.com/PabloFranco777/Proyecto-Estructuras.git>

