**BINARY TO GRAY CODE CONVERTER**

* **we are creating a simple C++ program using class to convert the binary number to gray code.**
* **The binary code will be given by user.**

**WHAT IS BINARY NUMBER?**

* **A binary number system is a number system that is used to represent various numbers using only two symbols “0” and “1”.**
* **It means in binary number only “0” and “1” will be used.**

**For binary to gray code conversion first we have to learn how to convert simple decimal number into binary number.**

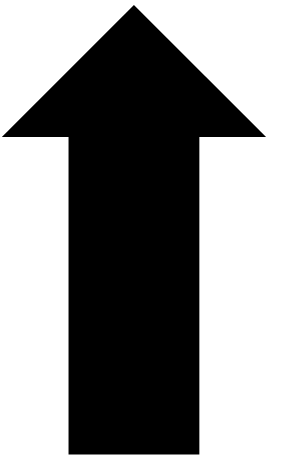
**For example:**

**We are converting decimal number 13 to binary number,**

**Binary number has base 2, so**

| **2** | **13** | **1** |
| --- | --- | --- |
| **2** | **6** | **0** |
| **2** | **3** | **1** |
|  | **1** | **1** |

**STEP 1: simply divide the number with 2 and write the remainder beside it.**



**STEP 2: now take the last remaining “1” to right side,**

**And note the number from down to up as shown above.**

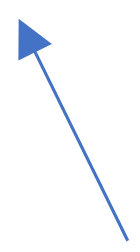
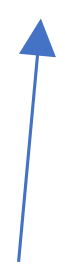
**Therefore, our binary number of 13 is 1101.**

**It can be written as**

**(13)10=(1101)2**

**CONVERTING BINARY TO GRAY CODE**

**For binary number there are some terminologies given that,**

**1 1 0 1**

**MSB LSB**

**Here;**

**MSB stands for most significant bit and**

**LSB stands for least significant bit.**

**Here are some steps to covert the binary number into gray code**

**STEP 1: The MSB of binary and gray code will always be the same during conversion.**

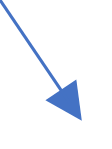
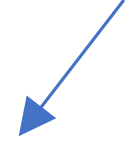
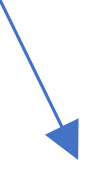
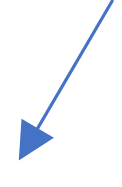
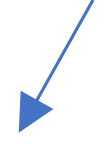
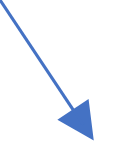
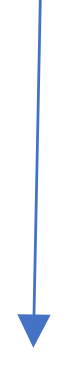
**STEP 2: after that we have to apply EX-OR logic between MSB and its neighbour bit to get the second number of gray code.**

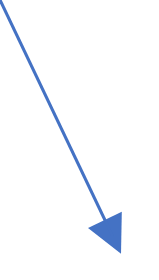
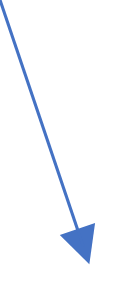
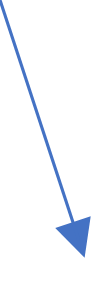
**STEP 3: after that we have to repeat the process until the completion of the number.**

**Let us take example,**

**Converting 1101 to gray code.**

**a3 a2 a1 a0**

**1 1 0 1**



**1 0 1 1**

**b3 b2 b1 b0**

**basically, we are performing a EX-OR logic;**

**so,**

**b3=a3**

**b2=a3** ⊕ **a2**

**b1=a2** ⊕ **a1**

**b0=a1** ⊕ **a0**

**for this we have to understand EX-OR logic**

* **When odd no. of “1” in input then output will be “1”.**

**Otherwise, output will be “0”.**

**Truth table for EX-OR logic:**

| **Input a** | **Input b** | **Output** |
| --- | --- | --- |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**Code:-**

**#include <iostream>**

**using namespace std;**

**class bin\_to\_gray{**

**public:**

**void logic(){**

**char bny[5];**

**cout<<"enter binary number ::"<<endl;**

**cin>>bny;**

**char gry[5];**

**int i=1;**

**while (i<5)**

**{**

**if(bny[i]==bny[i-1])**

**{**

**gry[i]='0';**

**}**

**else**

**{**

**gry[i]='1';**

**}**

**i++;**

**}**

**gry[0]=bny[0];**

**cout<<"gray code::"<<gry;**

**}**

**};**

**int main()**

**{**

**bin\_to\_gray x;**

**x.logic();**

**return 0;**

**}**