**Objective:**

* Handling file contents and exploring binary files.

**Challenge - X:** *Binary File conversion* **(**12**)**

Give us a human-readable text file (.txt) named as converted, by converting the binary file (.bin) containing the binary representation of ASCII characters.

**Binary (.bin) file format**

Binary files store information using sequences of 0s and 1s, unlike text files which use human-readable characters.

You are provided with .bin files that hold the binary codes for characters, encoded using the ASCII standard. Your task is to write a program that can:

1. **Read the binary data** from the .bin file.
2. **Interpret each byte** as the binary code for a character based on the ASCII standard.
3. **Convert the binary code** to its corresponding character.
4. **Write the recovered characters** to a new text file (.txt)

Consider the following binary file:

01001000 01100101 01101100 01101100 01101111

Text file after converting into human-readable form:

Hello

**1 byte** **ASCII value** **Character**

01001000 72 H

01100101 101 e

01101100 108 l

01101100 108 l

01101111 111 o

**Solution:**

#include <iostream>

#include <fstream>

using namespace std;

char binarytoChar(const char \* binary)

{

int character = 0;

int power = 1;

for (int i = 7; i >= 0; i--)

{

char bit = binary[i];

character += (bit - '0') \* power;

power \*= 2;

}

return (char)character;

}

int main()

{

ifstream binaryFile("Hello.bin", ios::binary);

ofstream textFile("converted.txt");

if (!binaryFile.is\_open() || !textFile.is\_open())

{

cout << "Error opening files!" << endl;

return 0;

}

char binaryData[9];

binaryData[8] = '\0';

while (binaryFile.read(binaryData, 8))

{

char space;

textFile << binarytoChar(binaryData);

binaryFile.read(&space, 1);

}

binaryFile.close();

textFile.close();

return 0;

}