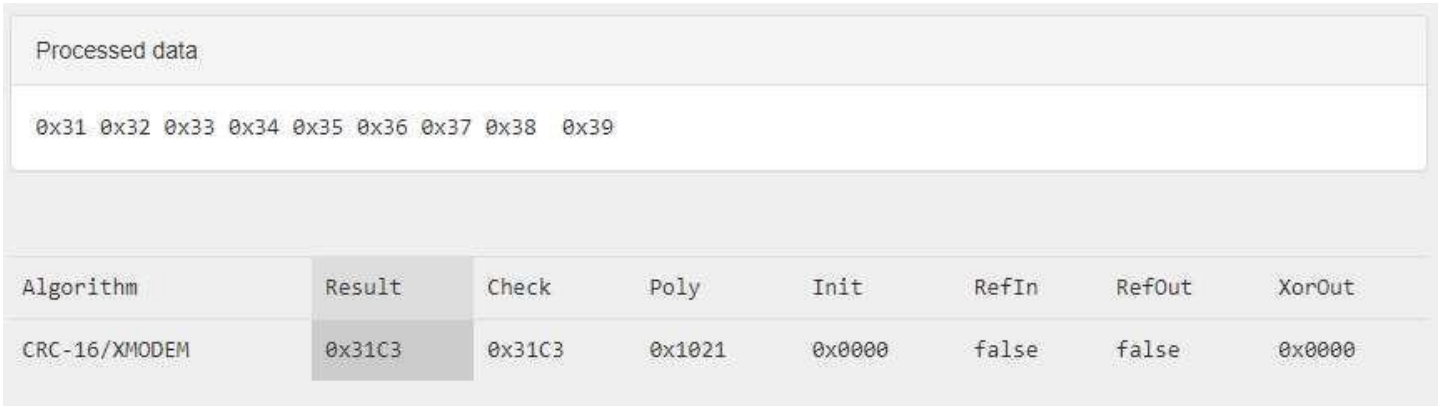
A small toy projector is used to project images. This projector has limited memory and can store only one image at a time. To update the image in the projector, it must be sent over a USB connection. However, since the image is large, it cannot be sent in a single packet. Therefore, the PC software divides the image into packets for transmission.

The packet frame looks like this

Header	Packet Length	Packet Number	Total Image Size	Payload	CRC16/ XMODEM	Footer
1 byte	1 byte	2 bytes	2 bytes	Payload size depends on packet length	2 bytes	1 byte

Each packet consists of the following components:

- **Header:** 1 byte with a fixed value of 0x7D, used to detect the beginning of the packet.
- **Packet Length:** 1 byte indicating the total number of bytes in the packet.
- **Packet Number:** 2 bytes (with the most significant byte sent first) representing the packet's order from 1 to N, where N is the total number of packets. The first packet of the image file will have a packet number of 1. The second packet of the image file will have a packet number of 2 and so on.
- **Total Image Size:** 2 bytes (with the most significant byte sent first) indicating the size of the image in bytes. For example, If the total image data is 300 bytes then the total image size field will be 300 in all the packets of the image which is being sent.
- **Payload:** The length can be calculated by subtracting other components from packet length. The payload contains the image data.
- **CRC-16/XMODEM:** 2 bytes (with the most significant byte sent first). A 16-bit checksum field using the CRC-16/XMODEM algorithm, as described on this website: Online CRC-8 CRC-16 CRC-32 Calculator. The checksum is calculated over Packet Length, Packet Number, Total Image size, and Payload. For



  Example, if CRC16/XMODEM is calculated over 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, then the calculated CRC is 0x31C3. The below screenshot is for your reference.

- **Footer:** 1 byte with a fixed value of 0xD7, used to detect the end of the packet.

Your task is to complete the imageReceiver function, which receives these packets, validates them, and saves the image in memory as the packets are received. **If any packet is dropped or invalid/corrupted** during transmission, the image should be erased, and all the remaining packets of that particular image should be ignored. This function is called whenever a packet is received.

The **eraseImageData** function should be called when the first image packet is received to erase the memory. If any packet is dropped or corrupted during transmission, you can call this function to erase the image.

The **saveImageData** function should be called every time when you want to save parts of the image data. Please note you are not allowed to edit the main function, **eraseImageData** function, **saveImageData** function.

Please complete the imageReceiver function, taking care to implement the necessary functionality. Also write a function to calculate crc16/xmodem of the packet as mentioned to make sure the packet is not corrupted.

The code is presented in the main.c file provided with this document. You can copy the program and paste it into the editor on https://www.onlinegdb.com/ or choose the upload file option on https://www.onlinegdb.com/ to open the main.c file directly in the editor. Make sure you choose the language C from the top right corner. When you run the program without editing, you will see "Toy Projector Firmware v1.0 is running" without any warnings and errors. You will also notice that the imageReceiver function is receiving a stream of image packets. If you write the imageReceiver function correctly, you will see the image printed in the consol.