```python
# Fixed constants
EXIT_SUCCESS = 0
EXIT_FAILURE = 1

BYTE_MASK = 0xFF
MAGIC_NO  = 0x497E  # required safeguard
TYPE      = 0x2     # required Type


class FileResponse():
    """
    Creating a file request.
    """

    def __init__(self, magicNum, statusCode, dataLength, _type=TYPE):
        """
        Init
        """
        self.magicNum = magicNum
        self.statusCode = statusCode
        self.dataLength = dataLength
        self._type = _type


    def encodeFixedHeader(self, record):
        """
        The Fixed Header is made up of 8 bytes. The Client
        sends these bytes over to the Server through the
        socket.
        - Stores byte informtion in a byte array.
        """
        # Encoding Fixed Header
        byte1 = self.magicNum >> 8
        byte2 = self.magicNum & BYTE_MASK
        byte3 = self._type
        byte4 = self.statusCode
        byte5 = self.dataLength >> 24
        byte6 = (self.dataLength >> 16) & BYTE_MASK
        byte7 = (self.dataLength >> 8) & BYTE_MASK
        byte8 = self.dataLength & BYTE_MASK

        record += (bytes([byte1]) + bytes([byte2]) + bytes([byte3]) + bytes([byte4]) +
                   bytes([byte5]) + bytes([byte6]) + bytes([byte7]) + bytes([byte8]))


    def responseChecker(self):
        """

        """
        if ((self.magicNum != MAGIC_NO) or (self._type != TYPE) or
            (self.statusCode != 1 and self.statusCode != 0)):
            return EXIT_FAILURE
        return EXIT_SUCCESS


def decodeFixedHeader(data):
    """
    Decodes the 8 byte Fixed Header and returns the three wanted
    values, (magicNum, _type and fileNameLen).
    """
    # Decoding Fixed Header
    magicNum = (data[0] << 8) | (data[1] & BYTE_MASK)
    _type = data[2]
    statusCode = data[3]
    dataLength = ((data[4] << 24) | ((data[5] << 16) & BYTE_MASK) |
                  ((data[6] << 8) & BYTE_MASK) | (data[7] & BYTE_MASK))

    return (magicNum, _type, statusCode, dataLength)
```