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

BYTE_MASK = 0xFF
MAGIC_NO  = 0x497E  # required safeguard
TYPE      = 0x1     # required Type


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

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

        # Set the length of a file name
        try:
            self.fileNameLen = len(fileName)  # got a type string
        except TypeError:
            self.fileNameLen = fileName       # got a type int


    def encodeFixedHeader(self, record):
        """
        The Fixed Header is made up of 5 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.fileNameLen >> 8
        byte5 = self.fileNameLen & BYTE_MASK

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


    def requestChecker(self):
        """
        Checks the validity of the File Request record and returns the
        status of the Fixed Header.
        - Returns 0 if record is correct.
        """
        checkFileLen = self.fileNameLen < 1 or self.fileNameLen > 2**10
        if (self.magicNum != MAGIC_NO) or checkFileLen or (self._type != TYPE):
            return EXIT_FAILURE
        return EXIT_SUCCESS


def decodeFixedHeader(data):
    """
    Decodes the 5 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]
    fileNameLen = (data[3] << 8) | (data[4] & BYTE_MASK)

    return (magicNum, _type, fileNameLen)
```