# 实验6.1

## 基本信息

* 实验名称：6.1 有噪信道编码的实现
* 实验类型：设计
* 实验地点：分散
* 实验时间：2周

## 目标

通过编写和操作一个重复码编码器，加深对有噪信道编码的理论知识与实际应用的理解。

## 内容

* 学习教师提供的demoBitString.py
  + 成功运行该演示程序，并理解所显示的内容的意义
  + 学习源代码，理解所有代码的原理、注释的含义
* 基于demoBitString.py，编写重复码编解码程序（及其单元测试），程序API如下：
* repetitionCoder.py encode LEN INPUT OUTPUT  
   LEN int, code length n, must be odd number and 2 < n < 10  
   INPUT path to the encoder input file  
   OUTPUT path to the encoder output file  
    
  repetitionCoder.py decode INPUT OUTPUT  
   INPUT path to the decoder input file  
   OUTPUT path to the decoder output file
  + encode 命令部分
    - LEN：重复码的码字长度，必须是大于2小于10的**奇数**
    - INPUT：输入文件（任意格式）
    - OUTPUT：输出文件（编码后的文件）
  + decode 命令部分
    - INPUT：输入文件（编码后的文件）
    - OUTPUT：译码后的文件

【注】encode命令的输出文件OUTPUT，decode命令的输入文件INPUT，都应具有以下的文件格式：

| **LEN** | **source length** | **codeword sequence** |
| --- | --- | --- |
| 1 byte | 4 bytes | multiple bytes |

* 编写误码率计算程序（及其单元测试），程序API如下：
* calcErrorRate.py INPUT1 INPUT2 RESULT  
   INPUT1 path to input file 1  
   INPUT2 path to input file 2  
   RESULT path to the result CSV file
  + INPUT1、INPUT2：进行相互比较的两个文件
  + RESULT：比较后计算结果所保存的CSV文件，格式如下： "INPUT1","INPUT2","error\_rate" 其中，error\_rate是按二进制位计算的误码率，即“错误位数/总位数”
* 操作
  + 选择任意一个大于10KB的文件，使用3种不同的重复码长度，分别生成编码后的文件。
    - 记是重复码长为的编码后文件。
  + 将每个，通过3个不同错误传递概率的BSC（其中一个）
    - 使用【实验3.1】的byteChannel及其他相关程序
    - 记是通过错误传递概率的BSC后的输出
  + 对每个，进行重复码解码
    - 记是解码后的输出
  + 对每个，计算与之间的误码率
    - 通过理论推导，得到“理论误码率”
    - 使用calcErrorRate程序，得到“实验误码率”
* 分析讨论
  + 从所得的“实验误码率”，总结各种不同参数的组合对信息传递的影响
  + 对比“理论误码率”和“实验误码率”，简要说明差距存在的原因
    - 【可选】如果实验误码率与理论值相差甚远，提出改进重复码编解码程序的思路，使其能接近理论值

## 提交

1. 程序压缩包，包含
   * 本实验中编写的程序源代码：代码文件本身，如.py文件。不包含其他文件非源代码文件，如开发工具的工程文件等。
   * **非Python/Java程序**，还需提供可执行程序：.exe文件。
   * 单元测试数据，以及数据的说明。
   * 程序使用说明：如运行环境、调用形式、参数的含义等。（实验报告中不需要复述相关内容。）
2. 实验压缩包（不需要包含任何程序相关文件）
   * 实验数据，以及数据的说明。
   * 实验报告：实验基本信息、目标、原理等标准内容。