第一题：

文件：

1. sample_sanger.fastq 为 FASTQ 格式

Fastq 格式是一种基于文本的存储生物序列和对应碱基（或氨基酸）质量的文件格式。最初由桑格研究所（Wellcome Trust Sanger Institute）开发出来，现已成为存储高通量测序数据的标准。

```
@HWI-ST466_135068617:8:1101:1438:36397/1
AGAGAACGTGGGCTTACATCAGACACTGGATCAGACACTAAACGAACTTAACTGTATATAAGCAAAACAGAAGAGTCTTGTTCCAACAGAAACTCTGGAG
+
CCCFFFFFDHHHHJJJJJJJJJH@GIIHIGJJJIIIJJJIJJJJIJJJIIJJJJJC@CGGHHGHJJCHHHHFFEFE>AEDECCDDEDDDD?C>?CDCCCCB
```

FASTQ 格式储存的序列信息，每 1 条 reads 的信息，可以分成 4 行。

第 1 行以“@”开头，后面跟着唯一的序列 ID 标识符，然后跟着可选的序列描述内容，标识符与描述内容用空格分开。

第 2 行就是测序得到的序列信息，一般用 ATCGN 来表示，其中 N 表示荧光信号干扰无法判断到底是哪个碱基。

第 3 行以“+”开始，可以储存一些附加信息，一般是空的。

第 4 行储存的是质量信息，与第 2 行的碱基序列是一一对应的，其中的每一个符号对应的 ASCII 值为 phred 值，可以简单理解为对应位置碱基的质量值，越大说明测序的质量越好。

问题：

1. sample_sanger.fastq 中，每条 reads 的第四行对应该序列的质量信息。质量值 Q 的计算方式为：对应字符的 ASCII 值减去 33。如第四行的某个位置字母为 C，那么该位置对应的 ASCII 为 67，对应的 Q 即为 67-33=34。ASCII 值可通过 ord()函数获取。

```
In [2]: ord('C')
Out[2]: 67
```

定义一个函数，输入序列质量信息，输出对应的 Q 值。并通过该函数将 sample_sanger.fastq 中每个位置的质量信息全部转为 Q 值，存入列表中，输出到 sample_sanger.fastq.txt.1。

```
@HWI-ST466_135068617:8:1101:1438:36397/1
AGAGAACGTGGGCTTACATCAGACACTGGATCAGACACTAAACGAACTTAACTGTATATAAGCAAAACAGAAGAGTCTTGTTCCAACAGAAACTCTGGAG
+
[67,67,67,70,70,70,70,70,68,72,72,72,72,74,74,74,74,74,74,74,74,72,64,71,73,73,72,73,71,74,74,74,73,73,73,74,74,74,73,74,74,74,74,73,74,74,74,73,73,74,74,74,74,74,67,64,67,71,71,72,72,71,72,74,74,67,72,72,72,72;
70,70,69,70,69,62,65,69,68,69,68,67,67,68,68,69,68,68,68,68,63,67,62,63,67,68,67,67,67,67,66]
@HWI-ST466_135068617:8:1101:1439:76914/1
GTTGGCTTTCGCCTGTAATCCCAACACTTTGGGAGGCCAGGACAGATCACTTGAGGTCAGGAGTTCGAGACCAGCCCAGCCAACATGGCGAAACCATGTC
+
[67,67,67,70,70,70,70,70,72,72,72,72,72,74,74,72,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,74,73,74,74,74,74,74,74,74,74,74,73,73,71,73,74,73,71,72,74,69,71,73,74,74,71,73,74,69,71,73,73,71,72,72,67,72,70,70,68,70,68,
68,68,68,68,68,68,68,68,68,68,68,68,66,68,68,68,68,68,68,68,68,68,68,68,68,68,68,68,68,68,67,68,65,67]
```

2. 对 sample_sanger.fastq 文件进行描述性统计：

1）读取该测序文件，获取 reads 总数并输出。

2）读取该文件中所有 reads 的质量值信息，计算每条序列的平均质量值，并输出平均质量值>30 的序列占总序列数的比例；

3）读取该文件中所有 reads 的质量值信息，计算每个位置的平均质量值并输出。假设一个文件中一共有 2 条 reads，read1 的第一个位置质量值为 20，read2 的第一个位置质量值为 30，那么该文件第一个位置的质量值平均值为 25。

4）读取该文件中所有 reads 的序列信息，计算每条序列的 GC 含量，输出 GC 含量<30%或者>70%的序列所占的比例。

3. 质控：过滤不符合条件的序列

1）读取每条 read 的序列信息，如果该序列中 GC 含量<30%或者>70%，则删除该 read；

2）读取每条 read 的序列信息，判断该序列是否重复出现过，如果重复出现过，则删除该 read；

3）截短序列：读取每条 read 的质量值信息，如果该序列的前 5 个碱基内有 Q 值<20 的，则对该序列进行截短操作。比如一条序列若第五位碱基 Q<20，那么则删除前五个碱基，保留其他剩余碱基及其质量值（序列长度缩小了 5）。若只有前四位 Q<20，则删除前四个。若前五个中只有第五个 Q<20，也要删掉前五个碱基，以此类推。该 read 对应的序列信息和质量值信息对应缩短。对所有 read 进行截短判断操作后，查看该序列 Q >20 的碱基的比例，若该比例小于 70%， 则删除该 read。

4.

数据稀疏化（downsampling）：为了减少冗余数据，随机选择原始数据中的 50%序列重新生成一个子集文件 sample_sanger_downsampled.fastq。稀疏化后的数据文件应保持原格式，并满足至少 70%的碱基位置的平均质量分数大于 30。

第二题：

文件：

1. GCA_000001405.15_GRCh38_no_alt_analysis_set.chr1.part.fa 为人 GRCh38 的一号染色体部分信息，FASTA 格式

2. codon.txt 为所有密码子信息文件

fasta 格式是一种基于文本用于表示核酸序列或多肽序列的格式。其中核酸或氨基酸均以单个字母来表示，且允许在序列前添加序列名及注释。

```
>chr1   AC:CM000663.2   gi:568336023   LN:248956422   rl:Chromosome   M5:6aef897c3d6ff0c78aff06ac189178dd   AS:GRCh38
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

FASTA 文件主要由两个部分构成序列头信息（有时包括一些其它的描述信息）和具体的序列数据头信息独占一行，以大于号（>）开头作为识别标记，其中除了记录该条序列的名字之外，有时候还会接上其它的信息。紧接的下一行是具体的序列内容，直到另一行碰到另一个大于号（>）开头的新序列或者文件末尾。测序得到的序列信息，一般用 ATCGN 来表示，其中 N 表示无法判断到底是哪个碱基。

转录操作不用考虑模板链问题，假设原始 DNA 序列位 ATC，输出 RNA 序列 UAG 即可。

问题：

1.将从第二行开始的序列信息连接成一行的完整序列，并删除所有序列中的 N。输出 chr1.part.txt.1。

2.定义一个函数 transcript，输入 DNA 序列，转录成为 mRNA 序列。并将 chr1.part.1 的 DNA 序列转录为 mRNA，输出 chr1.part.txt.2。

3.定义一个函数 translate，输入 mRNA，输出氨基酸序列（字符串）。将 chr1.part.txt.2 的 mRNA 翻译，每个氨基酸序列一行，输出 chr1.part.txt.3。

a)起始密码子为 AUG，从第一个起始密码子开始翻译，起始密码子也对应氨基酸，作为氨基酸序列第一位。如果没有起始密码子，则返回空字符串。

b)每三个碱基翻译成对应的氨基酸，遇到终止密码子停止。

c)循环上述操作直到读取完全部 mRNA 序列，输出最终的翻译的结果。

```
MRWQVVLFAAPFTPPNFPPSLHYALPGSGPWSRASASSALPPPQLVQKLGRSVLGLVPPI
```

4：密码子优化

问题：

某些生物中存在密码子偏好，即不同生物体对于同一种氨基酸，可能更偏向于使用某些密码子来进行编码。在同一条 DNA 序列中，有时可以通过密码子优化来增强表达效率。

1）请编写一个 find_optimal_codons 函数，读取 chr1.part.txt.2 和 codon.txt，根据翻译所在的序列统计翻译后氨基酸最频繁使用的密码子。比如氨基酸 A 对应的密码子有 4 个，统计这 4 个的使用数量，数量最高者为使用最频繁的密码子。

2）编写一个 optimize_mrna_sequence 函数将 chr1.part.txt.2 中的每个密码子替换为对应氨基酸最频繁使用的密码子。最终输出优化后的 mRNA 序列，并保存到 optimized_chr2.part.txt 文件。

注：find_optimal_codons 中只考虑起始密码子和终止密码子中间的氨基酸，而且 optimize_mrna_sequence 替换也应该只替换这些密码子，如果一个氨基酸最频繁的密码子不止一个，请选择 codon 文件中最靠前的，同时 find_optimal_codons 需要打印出每个氨基酸使用的最频繁的密码子。

第三题：

文件：

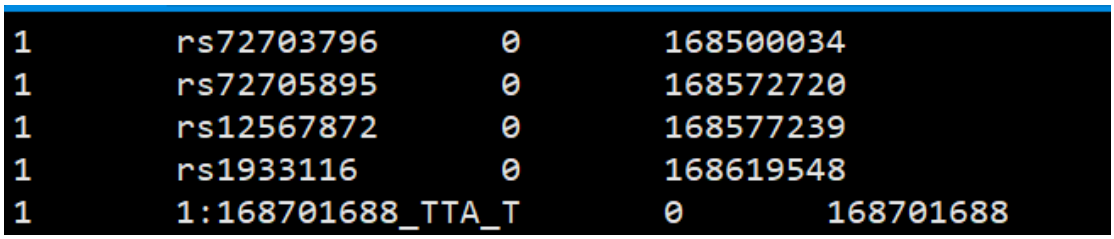
1.ukb_chr1_v3.filtered.remove_w46387_20200820.ped.part

2.ukb_chr1_v3.filtered.remove_w46387_20200820.map

ped 和 map 文件通常被用来存储群体的基因分型信息。其中，ped 格式是一个纯文本的文件，至少需要 6 列，每列有空格或者\t 分隔。这 6 列分别代表 Family ID，Individual ID，Paternal ID，Maternal ID，Sex（男性为 1，女性为 2），Phenotype。

剩余的列通常用来表示基因型信息，每个 snp 位点的基因型需要两列来表示，如图中所示，每个红框内为一个 SNP 的两个等位基因，该个体第一个 SNP 的基因型即为 AA。

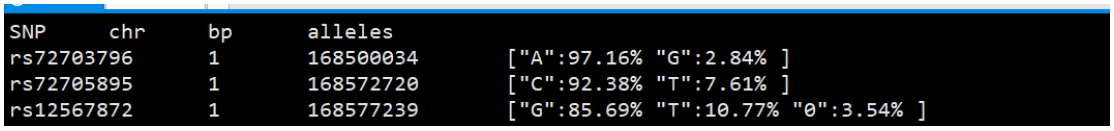
map 格式的文件，主要是 SNP 信息,四列分别是 CHR，SNPID,摩尔位置，物理位置。



问题：

1. 通过 map 文件，给 ped 文件第七列开始的 SNP 信息，按列加上 SNPID，去除 SNPID 不是以 rs 开头的 snp，将处理后的 map 和 ped 文件保存为 map.txt 和 ped.txt，并根据 snp 位点的基因型信息，统计计算每个 SNP 的各等位基因数量和所占比例。输出 ukb_chr1_v3.txt.1。

最终输出格式如图：



2.定义一个函数 maf，输入 SNPID,对单个 SNP 按照性别分别计算得到最小等等位基因频率，如果男女间最小等等位基因频率差值<0.05,则返回该 SNPID 以及两者的最小等等位基因频率。最小等位基因频率的定义：假设一个 SNP 有两个等位基因 A 和 T，统计 A 和 T 在群体中出现的概率，其中概率低的那个为最小等位基因，其概率为最小等位基因频率。

3.在同一条染色体上距离相近的 SNP 通常会一起遗传给后代。根据 map 文件 SNP 的物理位置，将 SNP 上下游 2KB 范围内的点视作处于同一连锁中。提取出 SNP 个数>3 个且<5 个的

连锁。并计算该连锁内所有 SNP 组成的单倍型及其概率。

比如某个连锁内有 4 个 SNP，其可能的基因型分别为（AA, AT，TT）和（GG, GT 和 TT），（CC,CT,TT）, (AA,AG,GG)。

那么该窗口的单倍型可能为 AGCA，AGCG,AGTA,AGTG,ATCA,ATCG,ATTA,ATTG,TGCA，TGCG,TGTA,TGTG,TTCA,TTCG,TTTA,TTTG 共 16 种。

根据 ped 文件输出这些可能的单倍型及其概率。输出 ukb_chr1_v3.txt.3。

4.检测隐性等位基因频率差异

问题： 某些疾病的遗传机制可能与隐性等位基因的频率有关。请编写一个函数来比较男性和女性群体中隐性等位基因的频率差异。假设隐性等位基因为两个相同的等位基因（如 AA 或 TT），统计每个 SNP 在男性和女性中隐性等位基因的频率，并输出频率差异 0.005 的 SNPID。

第四题：

文件：

1.uniprotkb_accession_Q6IEY1_2023_10_20.fasta

2.codon.txt 为所有密码子信息文件

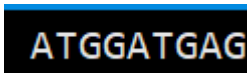
Fasta 格式是最常见的存储脱氧核糖核酸碱基序列或者是蛋白质氨基酸序列的文件格式，在 Fasta 中每一条 DNA 序列或者蛋白质序列都通过两行数据的形式进行存储。第一行一般描述了该条序列的名称（转录本、基因、各 RNA）以及来源（数据库、测序平台），还有可能会提供位置信息、数据的版本号，最重要的是这一行通常以“>”符号作为起始信息。然后第二行就承载了具体的序列信息。

问题：

1. 定义一个函数 rt，输入氨基酸，输出对应的所有可能的 DNA 三联碱基序列，如输入氨基酸 A，对应 mRNA 密码子为 GCG,GCA,GCC,GCU，则输出对应可能的 DNA 三联碱基序列为 CGC, CGT, CGG, CGA。

2. 读取 uniprotkb_accession_Q6IEY1_2023_10_20.fasta 中第一条氨基酸序列，通过上述函数 rt, 反推出该氨基酸序列**前 3 个**氨基酸所对应所有可能的 DNA 序列。

3. 定义一个函数 rtq, 输入氨基酸, 计算其对应的 DNA 序列的每个碱基位点 A、T、G、C 概率，并以 N 代替不确定碱基，返回 DNA 序列，如氨基酸 A 对应的返回值为 CGN。读取 uniprotkb_accession_Q6IEY1_2023_10_20.fasta，通过函数 rtq 返回每条多肽序列对应的 DNA 序列，输出到文件 uniprotkb_accession_Q6IEY1_2023_10_20.fasta.txt.2。



4. 定义一个函数 se,通过输入两个参数(搜索字段，匹配字符数）,输出该字段在该氨基酸序列中的出现位置和次数。如输入 ENH，2 两个参数，输出对应氨基酸序列中匹配‘E*H’，‘EN*’，‘*NH’字段的次数和所有索引位置。（模糊匹配，匹配 2 个字符），*指可以匹配任意单个字符

5. codon.txt 中每个氨基酸对应多个碱基序列，将这个数量看作该氨基酸的复杂分数。请设计一个函数 calculate_max_complexity_window，参数是窗口大小（自定义），返回氨基酸序列中平均复杂性最大的窗口的起始位置和该窗口的平均复杂性。

窗口滑动：假设序列为 MDGENHSV，自定义窗口大小为 3，第一个窗口为 MDG，第二个窗口为 DGE。以此类推。