README – DNA Sequence Analysis Script

Identifying Information

- **Programmer:** Puja Sanjay Thorat
- **Language:** Python 3.x
- **Date Submitted:** 09/29/30
- **Description:**
  This script downloads a specific DNA sequence (`chr1_GL383518v1_alt`) from the UCSC Genome Browser, processes it, and performs several tasks:
  1. Prints specific nucleotides from the sequence.
  2. Generates the reverse complement and prints selected bases.
  3. Counts nucleotide frequencies per kilobase and stores them in a nested dictionary.
  4. Converts the dictionary into lists, checks base count sums, and highlights discrepancies.

Objective

This assignment involves processing the DNA sequence of the chromosome `chr1_GL383518v1_alt` (GRCh38.p13) to:

1. Read and extract specific nucleotide positions.
2. Generate the reverse complement of the sequence.
3. Count nucleotide occurrences per kilobase.
4. Summarize nucleotide distributions across the entire chromosome.

Files Needed

1. **Main script**: `dna_sequence_analysis.py`.
2. **Data file (downloaded automatically):**
   - `chr1_GL383518v1_alt.fa.gz` → Compressed FASTA file downloaded from UCSC.
   - `chr1_GL383518v1_alt.fa` → Decompressed FASTA file used for analysis.

Required Libraries / Software

- **Python 3.x**
- Libraries:
    - `requests` → for downloading the sequence file.
    - `gzip` → for handling compressed files.
    - `shutil` → for file operations.

Install missing dependencies with:

```
pip install requests
```

Instructions for Running

1. **Download or copy** the script into a file named python assignment 2.py
2. Open a terminal or Colab notebook.
3. Run the script with: python assignment 2.py.
4. The script will:
    - Download the UCSC FASTA file.
    - Decompress it into a `.fa` file.
    - Process the sequence and print outputs step by step (Parts 1–4).

Files Created During Execution

1. **Downloaded file:**
    - `chr1_GL383518v1_alt.fa.gz` → compressed DNA sequence.
2. **Decompressed file:**
    - `chr1_GL383518v1_alt.fa` → DNA sequence used in analysis.
3. **Console outputs (not saved):**
    - Specific letters (10th, 758th, etc.)
    - Reverse complement outputs.
    - Dictionary counts and per-kilobase statistics.

Notes

- **Expected sum per kilobase:** 1000 bases
- **Possible discrepancies:**
    - The last kilobase has fewer than 1000 bases - 439 bases
    - Ambiguous nucleotides (`N`) are not counted in A/C/G/T

- **Environment:** Google Colab or any Python IDE with network access