Zanim rozpoczniesz realizację tego ćwiczenia przeczytaj całą instrukcję od początku do końca!

Do przygotowania pierwszej wersji rozwiązania należy posłużyć się dowolnym LLM. Możesz wybrać jeden z poniższych:

<https://chatgpt.com/>

<https://claude.ai/>

<https://www.deepseek.com/>

<https://copilot.microsoft.com/>

<https://github.com/features/copilot>

<https://huggingface.co/meta-llama>

<https://chat.qwen.ai/>

Zapoznaj się z opisem formatu FASTA, który używany jest w bioinformatyce:

<https://pl.wikipedia.org/wiki/FASTA_format>

**Z wybranym narzędziem AI utwórz w Pythonie** generator sekwencji nukleotydowych w formacie FASTA z następującymi wymaganiami minimalnymi (możesz wprowadzić więcej funkcjonalności jeśli chcesz):

1. Program powinien generować losową sekwencję DNA składającą się z nukleotydów (A, C, G, T)
2. Długość sekwencji powinna być określana przez użytkownika za pomocą funkcji input()
3. Program powinien zapytać użytkownika o nazwę (ID) sekwencji i jej opis
4. Wynik powinien być zapisany do pliku FASTA o nazwie odpowiadającej ID sekwencji z rozszerzeniem .fasta. W nagłówku w pliku FASTA powinno znaleźć się ID oraz opis podany przez użytkownika w ten sposób:

>{ID} {opis podany przez użytkownika}

{sekwencja DNA wygenerowana przez program}

1. Program powinien również wyświetlić statystyki sekwencji: procentową zawartość każdego nukleotydu oraz stosunek zawartości nukleotydów C i G względem A i T.
2. W losowym miejscu sekwencji program powinien wstawić twoje imię. Pamiętaj jednak żeby litery tworzące imię nie wpływały na statystyki sekwencji DNA oraz nie były liczone do jej długości.

**Wygląd interakcji z programem:**

Podaj długość sekwencji: 20

Podaj ID sekwencji: A123

Podaj opis sekwencji: Losowa sekwencja testowa

Podaj imię: Mike

Sekwencja została zapisana do pliku A123.fasta

Statystyki sekwencji:

A: 23.2%

C: 26.4%

G: 24.8%

T: 25.6%

%CG: 51.2

Przykładowo jak może wyglądać zawartość pliku (w tym przypadku plik będzie się nazywał A123.fasta):

”””

>A123 Losowa sekwencja testowa

ACTGCCTGAAMikeACGACTGCCT

”””

Zapisz sobie pierwszą wersję programu wygenerowaną przez AI i na niej już pracuj w pliku \*.py (tutaj nie używany notatników, proszę pracować w pliku py).

Wprowadź do programu minimum 3 ulepszenia w taki sposób, że kod oryginalny, który zmieniasz nadal pozostaje w programie, ale umieszczasz go w komentarzach. To ty decydujesz jakie ulepszenia wprowadzasz i dlaczego. W komentarzach w pliku \*.py umieść informację co zmieniasz na co i dlaczego zmieniasz kod wygenerowany przez LLM.

Możesz to zrobić w ten sposób:

# ORIGINAL:

# <stara wersja>

# MODIFIED (w nawiasie uzasadnienie dla zmiany):

<nowa wersja>

**Przygotuj finalną wersję pliku \*.py**, który będzie rozwiązaniem dla tego ćwiczenia, warunki brzegowe:

1. Plik powinien dać się uruchomić w środowisku programistycznym (np. VS Code, Thonny i podobne)
2. Umieść w formie komentarzy: cel programu i kontekst jego zastosowania
3. Umieść w formie komentarzy: szczegółowe objaśnienie działania każdej linijki kodu
4. Umieść w formie komentarzy zmiany, które wprowadzasz względem kodu wygenerowanego przez LLM według opisu powyżej
5. Dodaj inne elementy, które uważasz za ważne

**W terminie do 7 dni kalendarzowych** od dnia, w którym stacjonarnie odbywają się te ćwiczenia student powinien przygotować rozwiązanie projektu i umieścić plik **py** w oddzielnym folderze na portalu GitHub, a folder udostępnić prowadzącemu (użytkownik: Adv20202 jeśli prowadzącym jest Adam Kuzdraliński lub użytkownik tutorPJATK jeśli prowadzącą jest Anna Wąsowska).

Nazwa folderu: 2025py\_s11111

Nazwa pliku py: s11111\_2025.py

Oceniany będzie finalny kod znajdujący się w repozytorium. Historia commitów nie będzie brana pod uwagę. Jeżeli praca wykonywana była na różnych gałęziach to proszę je scalić do gałęzi głównej. Tylko główna gałąź będzie podlegała ocenie.

Przesłanie rozwiązania z niepoprawną nazwą pliku lub folderu traktowane jest jako brak przesłanego rozwiązania.