# **CEN0336 - Introdução a Programação de Computadores Aplicada a Ciências Biológicas**

# **Prova II**

Nome: Tiago Estevam Corrêa

No USP: 8969836

## **Instruções Gerais**

Vocês terão desde o dia 7 de dezembro as 18:00 e até o dia 21 de dezembro as 23:59 para resolver as atividades da prova.

São 4 perguntas/atividades, cada uma precisa de uma resposta neste arquivo de word, que tem que ser carregado no sistema e-disciplinas até o dia e horário limite. Além disso algumas perguntas precisam que vocês desenvolvam scripts e carregam num repositório github específico para a prova (indicado na pergunta 1). No dia 22 de dezembro as 00:01 o professor descarregará todos os repositórios para avaliar seus scripts. Os critérios de avaliação dos scripts são:

Eles têm que rodar sem erros de sintaxe (15%)

Eles têm que ser logicamente corretos (20%)

Eles têm que ter comentários explicando o que é feito em cada passo (20%)

Eles têm que ter controle da entrada dos usuários (10%)

Eles têm que ter controle de erros usando blocos try/except (20%)

Eles têm que implementar o uso de funções (15%)

## **Perguntas/Atividades**

1. Crie um repositório no github para esta prova. Escreva o endereço completo do seu repositório github como resposta desta pergunta.

Lembre-se que o repositório tem que ter o nome segundo o padrão:

P2\_NNNNNNN-CEN0336

Onde NNNNNNN é seu número USP.

A resposta tem que ter o endereço completo, iniciando com https://"

O professor irá clonar seu repositório, com as respostas para os outros pontos da prova.

Repositório: <https://github.com/TiagoEC/P2_8969836-CEN0336>

1. Crie um script de python com o nome script\_getORF.py que realize as seguintes operações:
2. Ler um arquivo multifasta da linha de comandos, usando o modulo sys.
3. Para cada registro fasta identificar o Open Reading Frame (Fase de Leitura Aberta: <https://pt.wikipedia.org/wiki/Fase_de_leitura_aberta> ou <https://en.wikipedia.org/wiki/Open_reading_frame>) mais comprido dentre as 6 fases de leitura possíveis. Os ORFs neste caso não têm introns. Repare que uma fase de leitura aberta tem que iniciar com um codon de início da tradução e terminar com um códon de terminação da tradução (<https://pt.wikipedia.org/wiki/C%C3%B3digo_gen%C3%A9tico>), é seu comprimento tem que ser múltiplo de três (3)
4. Daquele ORF mais comprido identificado, gerar o peptídeo codificado.
5. Imprimir o ORF mais comprido no arquivo ORF.fna. O formato da sequência tem que ser fasta, e seu identificador, tem que ser o identificador original, adicionando o string: “\_frameX\_START\_END”, onde X é a Fase de leitura (1,2,3,4,5 ou 6), e START e END são as coordenadas na sequencia
6. Imprimir o peptídeo correspondente ao ORF mais comprido no arquivo ORF.faa. O formato da sequência tem que ser fasta, e seu identificador, tem que ser o identificador original, adicionando o string: “\_frameX\_START\_END”, onde X é a Fase de leitura (1,2,3,4,5 ou 6), e START e END são as coordenadas na sequencia
7. Não esquece que tem que processar todos os registros do arquivo fasta original, i.e., tem que ter o mesmo número de entradas nos arquivos ORF.faa, ORF.fna que no arquivo original.
8. Carregue seu script, e os arquivos ORF.faa e ORF.fna no seu repositório.
9. Calcule o md5sum do seu script e coloque esse número como resposta desta pergunta.

Repositório:

<https://github.com/TiagoEC/P2_8969836-CEN0336/blob/main/script_getORF.py>

md5sum: a786d992568d473e631d7c210da6cc64

1. Explique a diferença entre os dois sets:

set1=set('accggttaata')  
set2={'accggttaata'}

O método set() cria um conjunto a partir de um iterável, então a partir de uma string, ele irá criar um conjunto com os caracteres únicos da string. Já a outra maneira de criar o set, pelo {}, será criado um conjunto com o item que está sendo fornecido, no caso será a string inteira.

1. Descreva com sus próprias palavras o que o seguinte pseudocódigo faz. Faz uma avaliação desse pseudocódigo e relate quaisquer problemas que possa ter.

Defina TOTAL com zero

Defina CONTADOR NOTAS como zero

While CONTADOR NOTAS é menor ou igual a dez

      Entre (input) a seguinte nota

      Some a nota ao TOTAL

Defina a média da disciplina como o TOTAL dividido por dez (TOTAL/10)

Imprima na tela a média da disciplina

Implemente esse pseudocódigo num script de python e carregue ele no seu repositório da prova com o nome notas.py. Calcule o md5sum do seu script e coloque esse número também como resposta desta pergunta.

Este pseudocódigo recebe 11 notas do usuário e calcula a média, dividindo por 10, e imprime a resposta.

Problemas:

No while, CONTADOR\_NOTAS nunca é atualizado e, portanto, o loop é infinito.

Como está contando do 0 ao 10 (inclusive), o número de notas recebidas está errado (11 ao invés de 10).

Na implementação literal do pseudocódigo, seria necessário transformar o input do usuário em int ou float para somar à nota.

Repositório:

<https://github.com/TiagoEC/P2_8969836-CEN0336/blob/main/notas.py>

md5sum: ed9c9d399bb406b5d0451e2ca8c1da8a