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

# **Prova II**

Nome: Ellen Grous

No USP: 11784980

## **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.

**Resposta**

<https://github.com/ellengrous/P2_11784980_CEN0336.git>

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.

**Resposta**

**md5sum:** 5d74e7e58ae2f2ce52154a68aa2bd9a02f23d35b5dac1e6c6e3fa7000d089b15

1. Explique a diferença entre os dois sets:

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

**Resposta**

Analisando a diferença entre:

set1=set('accggttaata')

set2={'accggttaata'}

Tem-se como resultado que o set 1 criará elementos únicos para cada caractere, ficando o resultado como: {'a', 'c', 't', 'g'}, uma vez que elimina os elementos duplicados. Enquanto que o set 2 criará um único conjunto com a string accggttaata {'accggttaata'} já que a função set não foi utilizada.

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.

**Resposta**

Explicação do pseudocódigo passo a passo:

1. Inicializa a variável TOTAL com zero;
2. Inicializa a variável CONTADOR NOTAS com zero;
3. Entra em um loop enquanto CONTADOR NOTAS for menor ou igual a dez;
4. Solicita ao usuário que insira uma nota;
5. Adiciona a nota à variável TOTAL;
6. Incrementa o CONTADOR NOTAS;
7. Após o loop, calcula a média da disciplina dividindo o TOTAL por dez;
8. Imprime a média da disciplina na tela.

Observação: Há um problema no código ao iterar sobre o loop, o mesmo não tem uma condição de parada, causando loop infinito no script. Adicionei na linha 22 a condição para que o contador seja incrementado. Dessa forma, o loop irá parar após a inserção de 10 notas.

O md5sum se encontra na próxima página.

**md5sum**: 80c7af36b341efb0c9173b5f2927605eeeec7e6349817125a3a456c28f2300e4