תרגיל בית 1 – פייתון וכלים ביו-אינפורמטיים:

שאלה 1:

ישנו ביולוג סינטטי משוגע שיצר גנום סינטטי אשר מכיל 2 נוקליאוטידים('b' , 'a') וכן 4 קודונים אשר מקודדים לחומצות אמינו לפי הפירוט בטבלה הבאה:

חומצה אמינית	קודון
X	aa
Y	ab
Z	ba
W	bb

כתבו פונקציה בשם **nt_2_aa(nt_vec)**.

- קלט: string של נוקליאוטידים.
- פלט: string של חומצות האמינו אליהם הנוקלאוטידים מקודדים.

הפונקציה צריכה לדעת להתמודד עם מקרה בו חלק מהנוקלאוטידים בקלט באותיות קטנות\ גדולות. במקרה בו אורך הרצף אינו תקין, יש **להחזיר** את המחרוזת "Invalid length".

דוגמת הרצה:

```
nt_seq1 = "aabbababbb"
aa_seq1 = nt_2_aa(nt_seq1)
print(aa_seq1)
    ➢ "XWYYW"

nt_seq2 = "aaBBabAbbb"
aa_seq2 = nt_2_aa(nt_seq2)
print(aa_seq2)
    ➢ "XWYYW"
```

שאלה 2:

*** בשאלה זו אין צורך לבדוק את תקינות הקלט***

מצורף הקובץ 'array.txt':

א. כתבו פונקציה בשם **read_array(file)** שקוראת את הקובץ כ-DataFrame.
   - קלט: שם הקובץ כ-string
   - פלט: משתנה מסוג DataFrame ששמות העמודות שלו הם שמות העמודות המופיעים בקובץ הטקסט.

- שימו לב כי בראש הקובץ יש כמה שורות המכילות טקסט שאינו חלק מה DataFrame- על הפונקציה לדעת להתמודד עם זה. הפונקציה תיבדק אך ורק על קובץ הטקסט המצורף.

ב. כתבו פונקציה בשם **find_gene_name (df, gene_name, column)** המוצאת את מספר השורה בה מופיע שם הגן.
   - קלטים:
     - df: מסוג DataFrame.
     - gene_name: מסוג string. שם הגן שנרצה למצוא ב-df.
     - column: מסוג string. שם העמודה בה נחפש את הגן.
   - פלט: מספר (int) המתאר את אינדקס השורה בה מופיע הגן.

הפונקציה תיבדק על ה-DataFrame שמתקבל כפלט מהפונקציה בסעיף א'.

ג. בסעיף זה נרצה למצוא את קואורדינטת ההתחלה והסיום של כל גן ב- DataFrame.
ידוע כי מערך הקואורדינטות עבור כל גן מיוצג כ- strings מהצורה הבאה: "x..y", כאשר x מייצג את קואורדינטת ההתחלה ו-y מייצג את קואורדינטת הסיום.
כתבו פונקציה בשם **create_coordinates_cols (df, column_name)** המוסיפה ל-DataFrame שהתקבל בסעיף א' שתי עמודות-
- עמודה בשם 'startCoordinate': עמודה זו תכיל את קואורדינטת ההתחלה של הגן.
- עמודה בשם 'endCoordinate': עמודה זו תכיל את קואורדינטת הסיום של הגן.
   - קלטים:
     - df: DataFrame שהתקבל בסעיף א'
     - column_name: שם העמודה שבה מופיעות הקואורדינטות.
   - פלט: DataFrame זהה לזה שהתקבל ככלט רק עם שתי העמודות הנוספות שתוארו לעיל. ערכי הקואורדינטות בעמודות החדשות צריכים להיות מסוג int.

ד. כתבו פונקציה בשם **drop_column (df, column)** שמורידה את עמודת הקואורדינטות המקורית.
   - קלטים:
     - df: ה-DataFrame שהתקבל בסעיף ג'
     - column: שם העמודה (string) שנרצה להוריד מה- DataFrame.
   - פלט: ה-DataFrame ללא העמודה column.

דוגמת הרצה:

```
# A
gene_df = read_array('array.txt')
print(gene_df.shape)
```

➢ (94,8)

```
print(gene_df.iloc[0])   # print the first row of the DataFrame
```

➢

Index	Location	Strand	Length	PID	Gene	Synonym	Code	COG
0	"1807..2169"	-	120	6319249	"PAU8"	"YAL068C"	-	-

```
# B
gene_row_idx = find_gene_name(gene_df, 'YAL067W-A', 'Synonym')
print(gene_row_idx)
```

➢ 1

```
# C
gene_df_with_coordinates = create_coordinates_cols(gene_df, "Location")
print(gene_df_with_coordinates.iloc[0])   # print the first row of the DataFrame
```

➢

Index	Location	Strand	Length	PID	Gene	Synonym	Code	COG	startCoordinate	endCoordinate
0	"1807..2169"	-	120	6319249	"PAU8"	"YAL068C"	-	-	1807	2169

```
# D
df_final = drop_column(gene_df_with_coordinates, "Location")
print(df_final.iloc[0])    # print the first row of the DataFrame
```

➢

Index	Strand	Length	PID	Gene	Synonym	Code	COG	startCoordinate	endCoordinate
0	-	120	6319249	"PAU8"	"YAL068C"	-	-	1807	2169

<u>שאלה 3:</u>

*** בשאלה זו אין צורך לבדוק את תקינות הקלט***

מצורף הקובץ "E_coli_ORF.csv". הקובץ מכיל רצפים של 4319 גנים של החיידק אי קולי (E.coli).

א. קראו את הקובץ- כתבו פונקציה בשם **read_orf(file)** המקבלת את שם הקובץ וקוראת אותו.
   - קלט: שם הקובץ (<u>string</u>).

- פלט: DataFrame המכיל עמודה בשם orf שערכיה הם רצפים מסוג string, ו-4319 שורות.

ב. חשבו את מספר ההופעות של כל קודון בכל אחד מהרצפים. כתבו פונקציה בשם **count_codons(sequences)**. המחשבת את מספר ההופעות של כל קודון בכל אחד מהרצפים בעזרת הפונקציה Counter שנלמדה בתרגול.
- קלט: עמודת הרצפים ב- DataFrame.
- פלט: רשימה (list) של אובייקטים מסוג Counter. אורך הרשימה הוא כמספר הרצפים (ואורך מספר השורות ב-DataFrame).

ג. מצאו את מספר הופעות הקודון המקסימלי בכל אחד מהגנים (ייתכן שבכל גן מדובר בקודון אחר). על מנת לעשות זאת, כתבו פונקציה בשם **find_highest_frequency_codon(codons_frequency)** :
- קלט: הרשימה (list) שהתקבלה מהפונקציה בסעיף ב'.
- פלט: רשימה (list) של מספר ההופעות המקסימלי (int) בכל אחד מהגנים. אורך הרשימה הוא כמספר הרצפים.

למשל- עבור הגן 'ATC ATC GGT GCA ATC' ברשימת הפלט ישמר המספר 3, מכיוון שהקודון ATC חוזר 3 פעמים והוא הקודון שחוזר הכי הרבה.

דוגמאות הרצה:

```
# A
orf_df = read_orf('E_cooli_ORF.csv')
print(orf_df.iloc[0])    # print the first row of the DataFrame
```

➢

Index	orf
0	"ATGAAACGCATTAGCACCACCA…"

```
# B
codons_frequency = count_codons(orf_df.orf)
print(codons_frequency[0])
```

➢ Counter({'ATG': 1,
'AAA': 1,
'CGC': 1,
'ATT': 3,
'AGC': 1,
'ACC': 7,
'ATC': 1,
'ACA': 1,
'GGT': 2,
'AAC': 1,
'GCG': 1,
'GGC': 1,
'TGA': 1})

```
# C
max_freq = find_highest_frequency_codon(codons_frequency)
print(max_freq[0])
```

➢ 7

שאלה 4:

*** בשאלה זו אין צורך לבדוק את תקינות הקלט***

מצורף הקובץ 'Yeast_gene.fa". הקובץ מכיל רצפים של 1000 גנים של שמר האפייה(S.cerevisiae) בפורמט FASTA. בשאלה זו יש להשתמש בספריה **SeqIO** כפי שנלמד בתרגול.

א. קראו את הקובץ- כתבו פונקציה בשם **read_fasta (file)** שמקבלת את שם הקובץ וקוראת אותו.
   - קלט: שם הקובץ (string).
   - פלט: DataFrame הכולל עמודה בשם header המכילה את שמות הגנים והערכים שלה הם מסוג string, ועמודה בשם sequence המכילה את הרצפים והערכים שלה הם מסוג Seq.

ב. כתבו פונקציה הנקראת **convert_to_rna(df)** המוסיפה ל- DataFrame עמודה שנקראת "rna_sequence". על העמודה להכיל את רצפי הרנ"א המתאימים לכל רצף דנ"א שבעמודה sequence.
   - קלט: ה- DataFrame שהתקבל בסעיף א'.
   - פלט: אותו DataFrame שהתקבל כקלט, רק עם עמודה נוספת הנקראת "rna_sequence". על רצפי הרנ"א להיות מסוג Seq, אותו סוג (type) של רצפי הדנ"א בעמודה sequence.

ג. כתבו פונקציה הנקראת **convert_to_aa(df)** המוסיפה ל- DataFrame עמודה שנקראת "aa_sequence". על העמודה להכיל את רצפי חומצות האמינו המתאימים לכל רצף דנ"א שבעמודה sequence.
   - קלט: ה- DataFrame שהתקבל בסעיף ב' (הכולל בתוכו את העמודה rna_sequence).
   - פלט: אותו DataFrame שהתקבל כקלט, רק עם עמודה נוספת הנקראת "aa_sequence". על רצפי חומצות האמינו להיות מסוג Seq, אותו סוג (type) של רצפי הדנ"א בעמודה sequence.

ד. כתבו פונקציה הנקראת **calc_seq_len(df)** המוסיפה ל- DataFrame עמודה שנקראת "sequence_len". על העמודה להכיל את האורך בקודונים של כל רצף דנ"א המופיע בעמודה sequence.
   - קלט: ה- DataFrame שהתקבל בסעיף ג' (הכולל בתוכו את העמודות של רצפי הרנ"א ורצפי חומצות האמינו).
   - פלט: אותו DataFrame שהתקבל כקלט, רק עם עמודה נוספת הנקראת "sequence_len". ערכי העמודה sequence_len צריכים להיות מסוג int.

ה. כעת נרצה לחשב את ה- GC content של רצף. ערך זה מחושב בצורה הבאה:

$$GC\ content(\%)=\left(\frac{N_G+N_C}{N_G+N_C+N_A+N_T}\right)\cdot 100$$

כתבו פונקציה שנקראת **GC_content_calc(seq)** המחשבת את ערך ה- GC של הרצף (%GC content).
   - קלט: רצף אחד מסוג Seq (זהה לסוג הרצפים בעמודה sequence).

- פלט: ערך ה- GC content (מסוג float) של רצף הקלט.

ו. כתבו פונקציה הנקראת **GC_content_total(df)** המוסיפה ל- DataFrame עמודה שנקראת "gc_content". על העמודה להכיל את ערכי ה- GC content של כל הרצפים ב- DataFrame שהתקבל בסעיפים הקודמים.
   - קלט: ה- DataFrame שהתקבל בסעיף ד' (הכולל בתוכו את העמודות שהתקבלו בסעיפים א'-ד').
   - פלט: אותו DataFrame שהתקבל כקלט, רק עם עמודה נוספת הנקראת "gc_content". ערכי העמודה gc_content צריכים להיות מסוג float.

דוגמאות הרצה:

# A
df = read_fasta("yeast_genes.fa")
print(df.shape)

- (1000,2)

print(df.iloc[0]) # print the first row of the DataFrame

-

Index	header	sequence
0	"YDL197C"	ATGCCAAAAAATCGTGGTGTCTTGGATG…

# B
df_w_rna = convert_to_rna(df)
print(df_w_rna.shape)

- (1000,3)

print(df_w_rna.iloc[0]) # print the first row of the DataFrame

-

Index	header	sequence	rna_sequence
0	"YDL197C"	ATGCCAAAAAATCGTGGTGTCTTGGATG…	AUGCCAAAAAAUCGUGGUGUCUUGGAUG…

# C
df_w_aa = convert_to_aa(df_w_rna)
print(df_w_aa.shape)

- (1000,4)

print(df_w_aa.iloc[0]) # print the first row of the DataFrame

-

Index	header	sequence	rna_sequence	aa_ sequence
0	"YDL197C"	ATGCCAAAAAATCGTGGTGTCT…	AUGCCAAAAAAUCGUGGUGUCU…	MPKNRGVLDAITRSVIDGSD…

# D
df_w_len = calc_seq_len(df_w_aa)
print(df_w_len.shape)

- (1000,5)

print(df_w_len.iloc[0]) # print the first row of the DataFrame

-

Index	header	sequence	rna_sequence	aa_ sequence	sequence_len
0	"YDL197C"	ATGCCAAAAAATCGTGG…	AUGCCAAAAAAUCGUGG…	MPKNRGVLDAITRSVIDGSD…	526

# E
s = Seq("ATCGGGTACG")
gc = GC_content_calc(s)
print(gc)

- 60.0

# F
df_gc_content = GC_content_total(df_with_len)
print(df_gc_content.iloc[0]) # print the first row of the DataFrame

-

Index	header	sequence	rna_sequence	aa_ sequence	sequence_len	gc_content
0	"YDL197C"	ATGCCAAAAAAT…	AUGCCAAAAAAU…	MPKNRGVLDAITRSV…	526	43.2192648922687