

Lab Assignment-4.1

Name: M. Hasini

Ht.no: 2303A51109

Batch: 02

Q1. Zero-Shot Prompting (Basic Lab Task)

Task:

Write a Python function that classifies a given text as Spam or Not Spam using zero-shot prompting.

Steps:

1. Construct a prompt without any examples.
2. Clearly specify the output labels.
3. Display only the predicted label.

Input:

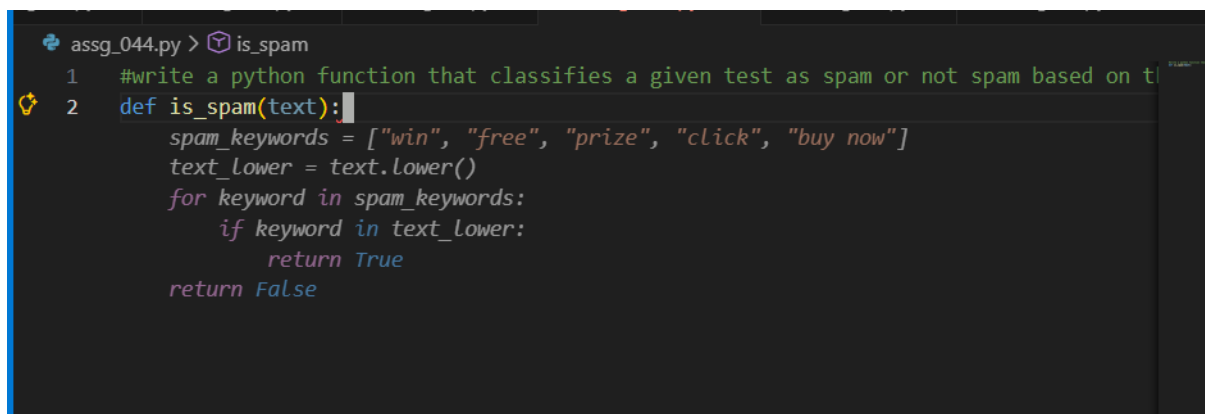
"Congratulations! You have won a free lottery ticket."

Expected Output:

Spam

Prompt:

#write a python function that classifies a given test as spam or not spam based on the presence of certain keywords



```
assg_044.py > is_spam
1 #write a python function that classifies a given test as spam or not spam based on the
2 def is_spam(text):
    spam_keywords = ["win", "free", "prize", "click", "buy now"]
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return True
    return False
```

```
assg_044.py > ...
Editors #write a python function that classifies a given text as spam or not spam based o
2 def is_spam(text):
3     spam_keywords = ["win", "free", "prize", "click", "buy now"]
4     text_lower = text.lower()
5     for keyword in spam_keywords:
6         if keyword in text_lower:
7             return True
8     return False
9 user_input = input("Enter the text to classify: ")
10 if is_spam(user_input):
11     print("The text is classified as spam.")
12 else:
13     print("The text is not spam.")
```

Code:

```
def is_spam(text):

    spam_keywords = ["win", "free", "prize", "click", "buy now"]

    text_lower = text.lower()

    for keyword in spam_keywords:

        if keyword in text_lower:

            return True

    return False

user_input = input("Enter the text to classify: ")

if is_spam(user_input):

    print("The text is classified as spam.")

else:

    print("The text is not spam.")
```

input/output:

```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the text to classify: hello,click the link below to claim the prizes
The text is classified as spam.
PS C:\Users\hasin>
```

Q2. One-Shot Prompting (Emotion detection)

Task:

Write a Python program that detects the emotion of a sentence using one-shot prompting.

Emotions: ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']

Steps:

1. Provide one labeled example inside the prompt.
2. Take a sentence as input.
3. Print the predicted emotion

Prompt:

emotions: ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']

write a python program that detect the mood of the person and take the sentence from the user if any word from the emotions list is present in the sentence then display that emotion otherwise display no emotion detected.

```
15 """
16 emotions: ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
17 write a python program that detects the emotion of sentence using the above list of
18 """
19 """emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
20 user_emotion = input("Enter an emotion to check: ").strip().lower()
21 if user_emotion in emotions:
22     print(f"The emotion '{user_emotion}' is present in the list.")
23 else:
24     print(f"The emotion '{user_emotion}' is not present in the list.")"""
```

```
15 """
16 emotions: ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
17 write a python program that detect the mood of the person and take the sentence from
18 """
19 emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
20 user_input = input("Enter a sentence to detect emotion: ").lower()
21 detected_emotions = [emotion for emotion in emotions if emotion in user_input]
22 if detected_emotions:
23     print(f"Detected emotions: {' '.join(detected_emotions)}")
24 else:
25     print("No emotion detected.")
26
```

Code:

```
emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
```

```
user_input = input("Enter a sentence to detect emotion: ").lower()
```

```
detected_emotions = [emotion for emotion in emotions if emotion in user_input]
```

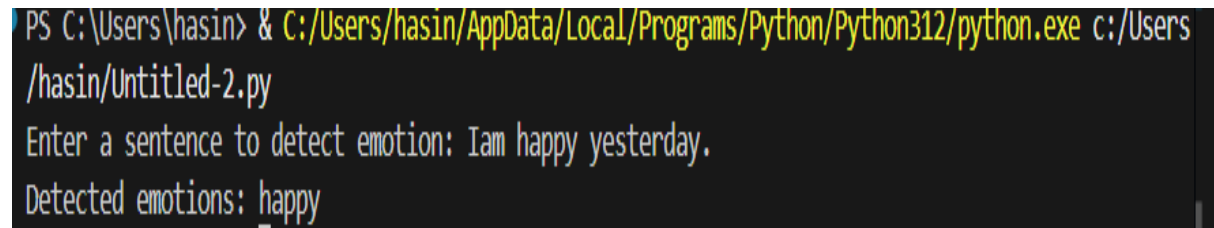
```
if detected_emotions:
```

```
    print(f'Detected emotions: {' '.join(detected_emotions)}')
```

```
else:
```

```
    print("No emotion detected.")
```

input/output:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter a sentence to detect emotion: Iam happy yesterday.
Detected emotions: happy
```

Q3. Few-Shot Prompting (Student Grading Based on Marks)

Task:

Write a Python program that predicts a student's grade based on marks using few-shot prompting.

Grades:

['A', 'B', 'C', 'D', 'F']

Grading Criteria (to be inferred from examples):

- 90–100 → A
- 80–89 → B
- 70–79 → C
- 60–69 → D
- Below 60 → F

Prompt:

90-100=A

80-89=B

70-79=C

60-69=D

below 60=F

write a python program that predicts a student's grade based on marks obtained and only accept positive integer values from the user otherwise display invalid input.

```

32 below 60=F
33 write a python program that predicts a student's grade based on marks obtained and o
34 """
35 """try:
    marks = int(input("Enter the marks (0-100): "))
    if marks < 0 or marks > 100:
        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= marks <= 100:
            grade = 'A'
        elif 80 <= marks <= 89:
            grade = 'B'
        elif 70 <= marks <= 79:
            grade = 'C'
        elif 60 <= marks <= 69:
            grade = 'D'
        else:
            grade = 'F'
        print(f"The grade for {marks} is: {grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")"""
36

```

```

27 """
28 90-100=A
29 80-89=B
30 70-79=C
31 60-69=D
32 below 60=F
33 write a python program that predicts a student's grade based on marks obtained and o
34 """
35 try:
36     marks = int(input("Enter the marks (0-100): "))
37     if marks < 0 or marks > 100:
38         print("Invalid input. Please enter a positive integer between 0 and 100.")
39     else:
40         if 90 <= marks <= 100:
41             grade = 'A'
42         elif 80 <= marks <= 89:
43             grade = 'B'
44         elif 70 <= marks <= 79:
45             grade = 'C'
46         elif 60 <= marks <= 69:
47             grade = 'D'
48         else:
49             grade = 'F'
50         print(f"The grade for {marks} is: {grade}")
51 except ValueError:
52     print("Invalid input. Please enter a valid integer.")
53

```

Code:

try:

```
marks = int(input("Enter the marks (0-100): "))
```

```
if marks < 0 or marks > 100:
```

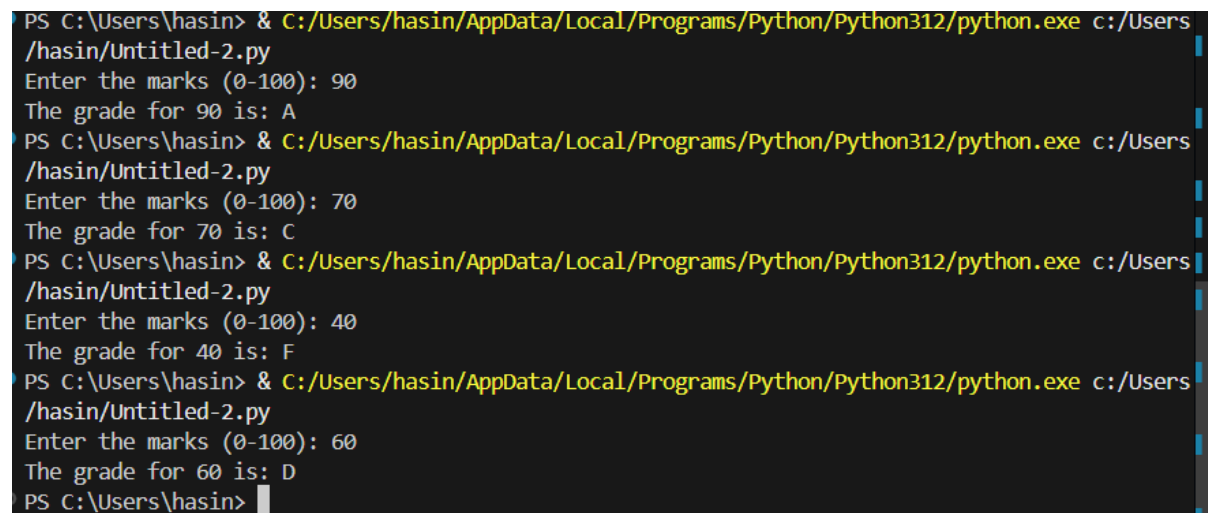
```

        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= marks <= 100:
            grade = 'A'
        elif 80 <= marks <= 89:
            grade = 'B'
        elif 70 <= marks <= 79:
            grade = 'C'
        elif 60 <= marks <= 69:
            grade = 'D'
        else:
            grade = 'F'

    print(f"The grade for {marks} is: {grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")

```

Inputs/outputs:



```

PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks (0-100): 90
The grade for 90 is: A
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks (0-100): 70
The grade for 70 is: C
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks (0-100): 40
The grade for 40 is: F
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks (0-100): 60
The grade for 60 is: D
PS C:\Users\hasin>

```

Q4. Multi-Shot Prompting (Indian Zodiac Sign Prediction using Month Name)

Task:

Write a Python program that predicts a person's Indian Zodiac sign (Rashi) based on the month of birth (month name) using multi-shot prompting.

Indian Zodiac Order (Simplified Month-Based Model): The Indian Zodiac cycle starts in March with Mesha and follows this order:

March → Mesha
April → Vrishabha
May → Mithuna
June → Karka
July → Simha
August → Kanya
September → Tula
October → Vrischika
November → Dhanu
December → Makara
January → Kumbha
February → Meena

Prompt:

march=mesha

april=vrishabha

may=mithuna

june=karka

july=simha

august=kanya

september=tula

october=vrischika

november=dhanus

december=makara

january=kumbha

february=meena

write a python code to accept month from the user and display the corresponding zodiac sign and only accept valid month names otherwise display invalid input

```

54
55
56 """
57 march=mesha
58 april=vrishabha
59 may=mithuna
60 june=karka
61 july=simha
62 august=kanya
63 september=tula
64 october=vrischika
65 november=dhanus
66 december=makara
67 january=kumbha
68 february=meena
69 write a python code to accept month from the user and display the corresponding zodiac sign
70
71 """
72 month_to_zodiac = {
73     "march": "mesha",
74     "april": "vrishabha",
75     "may": "mithuna",
76     "june": "karka",
77     "july": "simha",
78     "august": "kanya",
79     "september": "tula",
80     "october": "vrischika",
81     "november": "dhanus",
82     "december": "makara",
83     "january": "kumbha",
84     "february": "meena"
85 }
86 user_month = input("Enter a month: ").strip().lower()
87 zodiac_sign = month_to_zodiac.get(user_month)
88 if zodiac_sign:
89     print(f"The zodiac sign for {user_month.capitalize()} is {zodiac_sign}.")
90 else:
91     print("Invalid input. Please enter a valid month name.")
92

```

Code:

```

month_to_zodiac = {
    "march": "mesha",
    "april": "vrishabha",
    "may": "mithuna",
    "june": "karka",
    "july": "simha",
    "august": "kanya",

```



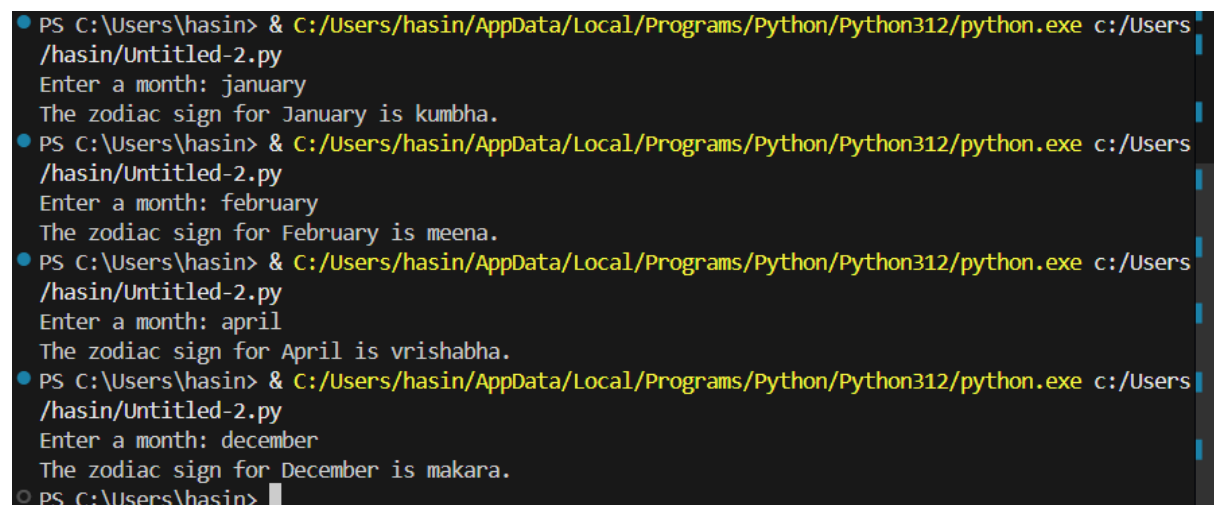
```

    "september": "tula",
    "october": "vrischika",
    "november": "dhanus",
    "december": "makara",
    "january": "kumbha",
    "february": "meena"
}

user_month = input("Enter a month: ").strip().lower()
zodiac_sign = month_to_zodiac.get(user_month)
if zodiac_sign:
    print(f"The zodiac sign for {user_month.capitalize()} is {zodiac_sign}.")
else:
    print("Invalid input. Please enter a valid month name.")

```

Inputs/outputs:



```

PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter a month: january
The zodiac sign for January is kumbha.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter a month: february
The zodiac sign for February is meena.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter a month: april
The zodiac sign for April is vrishabha.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter a month: december
The zodiac sign for December is makara.
PS C:\Users\hasin>

```

Q5. Result Analysis Based on Marks

Task: Write a Python program that determines whether a student Passes or Fails based on marks using Chain-of-Thought (CoT) prompting.

Result Categories:

['Pass', 'Fail']

Prompt:

take marks as an input from the user

if marks is greater than or equal to 40 then print pass

if marks is less than 40 then print fail

```
93  """
94  take marks as an input from the user
95  if marks is greater than or equal to 40 then print pass
96  if marks is less than 40 then print fail
97
98  """
99  """try:
    marks = int(input("Enter the marks: "))
    if marks < 0:
        print("Invalid input. Please enter a positive integer for marks.")
    else:
        if marks >= 40:
            print("Pass")
        else:
            print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer for marks.")"""
100  ⚡
```

```
"""
take marks as an input from the user
if marks is greater than or equal to 40 then print pass
if marks is less than 40 then print fail

"""
try:
    marks = int(input("Enter the marks: "))
    if marks < 0:
        print("Invalid input. Please enter a positive integer.")
    else:
        if marks >= 40:
            print("Pass")
        else:
            print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Code:

try:

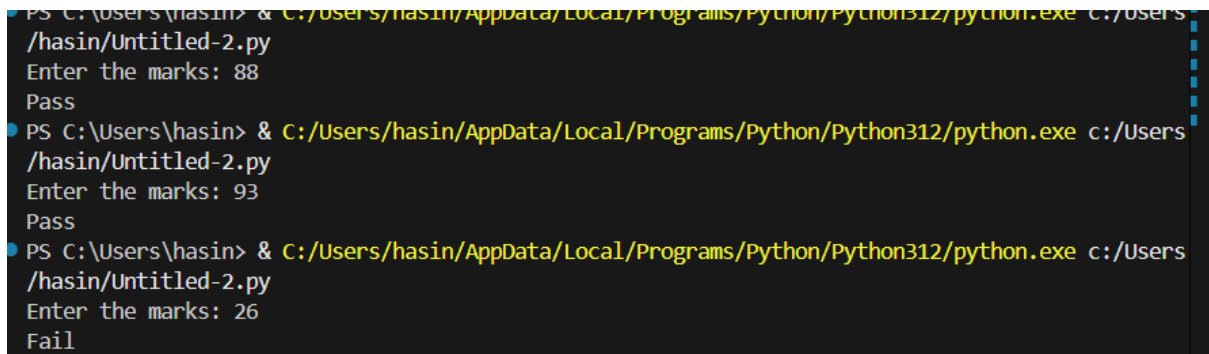
```
marks = int(input("Enter the marks: "))
```

```

if marks < 0:
    print("Invalid input. Please enter a positive integer.")
else:
    if marks >= 40:
        print("Pass")
    else:
        print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer.")

```

Inputs/outputs:



```

PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks: 88
Pass
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks: 93
Pass
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter the marks: 26
Fail

```

Q6 Voting Eligibility Check (Chain-of-Thought Prompting)

Task: Write a Python program that determines whether a person is eligible to vote using Chain-of-Thought (CoT) prompting.

Prompt:

take age from the user

if age is equal or greater than 18

then print eligible to vote

if age is less than 18

print not eligible to vote

prompt:

take age from the user

if age is equal or greater than 18

then print eligible to vote

if age is less than 18

print not eligible to vote

```

111  """
112  take age from the user
113  if age is equal or greater than 18
114  then print eligible to vote
115  if age is less than 18
116  print not eligible to vote
117  """
118  """try:
119      age = int(input("Enter your age: "))
          if age < 0:
              print("Invalid input. Please enter a positive integer.")
          else:
              if age >= 18:
                  print("Eligible to vote")
              else:
                  print("Not eligible to vote")
      except ValueError:
          print("Invalid input. Please enter a valid integer.")"""
120

```

```

111  """
112  take age from the user
113  if age is equal or greater than 18
114  then print eligible to vote
115  if age is less than 18
116  print not eligible to vote
117  """
118  try:
119      age = int(input("Enter your age: "))
120      if age < 0:
121          print("Invalid input. Please enter a positive integer.")
122      else:
123          if age >= 18:
124              print("Eligible to vote")
125          else:
126              print("Not eligible to vote")
127  except ValueError:
128      print("Invalid input. Please enter a valid integer.")
129
130

```

Code:

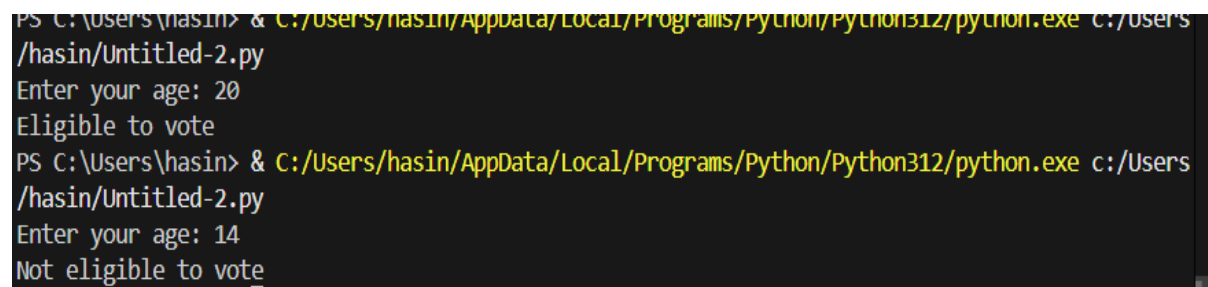
try:

```
age = int(input("Enter your age: "))  
if age < 0:  
    print("Invalid input. Please enter a positive integer.")  
else:  
    if age >= 18:  
        print("Eligible to vote")  
    else:  
        print("Not eligible to vote")
```

except ValueError:

```
print("Invalid input. Please enter a valid integer.")
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users  
/hasin/Untitled-2.py  
Enter your age: 20  
Eligible to vote  
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users  
/hasin/Untitled-2.py  
Enter your age: 14  
Not eligible to vote
```

Q7 Prompt Chaining (String Processing – Palindrome Names)

Task: Write a Python program that uses the prompt chaining technique to identify palindrome names from a list of student names.

Prompt:

take list of names form the user

if student names raa palindrome

then print those names in the form of list

```

130 """
131 take list of names from the user
132 if student names are palindrome
133 then print those names in the form of list
134
135 """
136 def is_palindrome(name):
137
138

```

```

139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Code:

```

def is_palindrome(name):
    return name == name[::-1]

students = input("Enter student names separated by commas: ").split(",")
palindrome_students = [name.strip() for name in students if is_palindrome(name.strip())]
print("Palindrome names:", palindrome_students)

```

Inputs/outputs:

```

PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Enter student names separated by commas: hasini,bob,anvitha,anjali
C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Palindrome names: ['bob']

```

Q8 Prompt Chaining (String Processing – Word Length Analysis)

Task: Write a Python program that uses **prompt chaining** to analyze a list of words. In the first prompt, generate a list of words. In the second prompt, traverse the list and calculate the length of each word. In the third prompt, use the output of the previous step to determine whether each word is **Short** (length less than 5) or **Long** (length greater than or equal to 5), and display the result for each word

Prompt:

take list of words from the user

if the length of the individual word is greater than 5 then the the word is longs word

else the word is short

print the longs words and short words in the form of list

```
142 """
143     take list of words from the user
144     if the length of the individual word is greater than 5 then the the word is longs word
145     else the word is short
146     print the longs words and short words in the form of list
147 """
148 words = input("Enter words separated by commas: ").split(",")
149 Long_words = [word.strip() for word in words if len(word.strip()) > 5]
150
151
```

```
142 """
143 take list of words from the user
144 if the length of the individual word is greater than 5 then the the word is longs word
145 else the word is short
146 print the longs words and short words in the form of list
147 """
148 words = input("Enter words separated by commas: ").split(",")
149 long_words = [word.strip() for word in words if len(word.strip()) > 5]
150 short_words = [word.strip() for word in words if len(word.strip()) <= 5]
151 print("Long words:", long_words)
152 print("Short words:", short_words)
153 |
154
```

Inputs/outputs:

```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/hasin/Untitled-2.py
Enter words separated by commas: eagle,jackle,lion
Long words: ['jackle']
Short words: ['eagle', 'lion']
```