**Programming Exercise 2**

**Name:** Dylan Hamid

**Date Created:** 6/8/2025

**Program Description:**

This program checks if an email message is spam. The program scans an input message for 30 keywords, and displays a spam score, the phrases found in the message, and an evaluation on if the message is spam.

**Functions used in the Program (list in order as they are called):**

1. **Function Name:** main()

**Description:** Prompts user for email message, and displays analyzed information (scam score, phrases found, judgement on if message is scam).

**Parameters:** none

**Variables:**

* email\_message(string): holds the email message input by user.
* return\_values(list): return value from spam\_checker(). Multi-dimensional list holding the spam score and the phrases found in message.
* score(int): Hold’s spam score as its own variable.
* Phrase\_list(list): List holding the phrases found in the message.

**Logical Steps:**

1. Prompts user for email message
2. Calls spam\_checker()
3. Assigns values from spam\_checker() to their own variables
4. Prints values from spam\_checker()
5. Checks spam score, and uses it to evaluate whether the message is likely to be spam.

**Returns:** none

2. **Function Name:** spam\_checker()

**Description:** Iterates over a word bank to check for common spam phrases in email message.

**Parameters:**

* email(string): email\_message from main(). Holds the email message input by user

**Variables:**

* email(string): Holds the email message input by user.
* Spam\_score(int): Accumulator. Accumulates for each occurrence of each phrase found in message
* Phrase\_list(list): Is appended when a phrase is found in the message.
* Spam\_phrase(list): word bank for common spam phrases.
* Occurrence\_list(list): list containing occurrence of phrases found in email messages.
* Return\_values(list): List containing spam\_score and occurrence\_list, to be returned to main()

**Logical Steps:**

1. Initializes email, spam\_score, phrase\_list, spam\_phrases
2. Loops over each phrase in spam\_phrases
   1. Finds all occurrence of each phrase in message, assigns to occurrence\_list
   2. Accumulates spam\_score for each item in occurrence\_list
   3. Appends current phrase to phrase\_list if occurrence\_list > 0
3. Assigns spam\_score and phrase\_list to return\_values, returns to main()

**Returns:** return\_values. The spam score and the phrases found in email message.

**Logical Steps:**

1. Call main()
2. Main() calls spam\_checker

**Link to your repository:** https://github.com/dyhlan/COP2373

**Output Screenshot: (make sure big enough so I can see)**

