Investigation:

The email contains exactly one ‘@’ symbol.

The ‘@’ symbol is not at the start or end of the email.

‘@’ should not be in the last six places of the email.

The prefix (part before ‘@’ is alphanumeric, does not start with a digit, and is less than 64 characters long.

The domain (part after ‘@’) is less than 255 characters long and contains at least one dot.

There are no consecutive dots in prefix or domain.

The email does not contain a comma.

The domain is not entirely numeric.

Numbers and Underscores are not allowed at the starting of the prefix.

Special characters (e.g. #,$,%,\*,e.t.c.) are not allowed in email address.

Underscores can be used in email address in prefix but not at first position.

Email should end with a valid domain (e.g. .com, .net, .in, .edu, e.t.c.).

1. What is the length of the string?

Length of the string in python can be found by using len() function.

In the current minor project, the local part should be less than 64 characters

And the domain part should be less than 255 characters.

I used it in the email validation program to check the length of local part and domain

part.

1. Is a specified character in the string?

In python we can use membership operator to check if a specified character is in

the string.

I used it in the email validation program to check the presence of ‘@’ in the given email.

1. What character is at a specific location in the string?

We use indexing to find a character at a specific location in string.

In the current email validation program we use indexing to make sure that first and last position of email should not be ‘@’ symbol.

1. What is the location in the string of a specific character?

In python we can find location of a specific character in string using

Variablename.index(“requiredcharacter”) . printing this will give index of

Required character.

In current email validation program we use this to find the location of ‘@’

symbol to check if it is not in the last six places of the email.

1. How many of a specific character are in the string?

To find how many times a specific character is repeated in a string, we use

string.count(“required character”).

In current email validation program we use this to count number of dots in domain

part to check it has at least one dot in it.

1. How can a string be split into smaller sub-strings?

A string in python can be split using split() function.

Syntax:- string.split(separator,max\_limit)

In current email validation program we use this function to split the given email

address into local part and domain part with “@” as seperation.

PSEUDOCODE:

Procedure validate\_email(line, outfile):

Convert line to lowercase

If '@' is in line:

If count of '@' in line is 1:

Split line at '@' into components

Assign prefix and domain from components

For each character in line:

Find the index of '@'

Calculate the distance from the end of the line to '@'

If prefix and domain exist and prefix doesn't start or end with '@' and the distance from the end is less than -6:

If prefix (with '\_' removed) is alphabetic or alphanumeric, and prefix doesn't start with a digit, and length of prefix is less than 64, and length of domain is less than 255, and domain contains at least one '.':

If '..' is not in prefix and domain, and ',' is not in line, and domain (with '.' removed) is not numeric, and domain (with '.' removed) is alphanumeric or alphabetic or contains '-':

Define valid top-level domains (TLDs)

If domain ends with any of the valid TLDs:

Write "\*\*\*VALID\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID (Invalid TLD)\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID\*\*\* {line}" to outfile

Else:

Write "\*\*\*INVALID\*\*\* {line}" to outfile

Define file\_path and output\_path

Open file\_path and output\_path

Read lines from file

For each line in lines:

Call validate\_email(line, outfile)

Prompt user to enter an email address

Open output\_path

Call validate\_email(email, outfile)

Close input file

Close output file