How to Use:

Enter a url into the input space when prompted.

That is all, the program with parse the url and return the components or errors found.

You may rerun the program to fix any url errors.

The program parses systematically through the url to match components or generate readable

and correctable errors.

Possible Errors:

1. Protocol

1. Cannot parse for protocol ("://" not found)

2. Protocol [input] is invalid - must be "http","https", or "ftp"

2. Domain

1. Cannot parse for domain (Port or Filepath beginning not found)

2. Domain format does not match: must be in form 'x.y.z'

3. Filepath beginning not found, check domain formatting (if no port)

4. Host [input] is invalid - must match "com", "net", "edu", "biz", or "gov"

3. Port - optional

1. Port number must be between 1 and 65535

2. Filepath beginning not found, check port formatting

4. Filepath

1. Filepath not found (check the beginning and ends of input)

2. Filepath is invalid - must end with ".html" or ".htm"

5. Parameters - optional

1. Query must follow format of "key=value"

2. Multiple parameters found invalid - must join by "&"

3. Parameters expected: must begin with "?" after filepath

Order:

USER CALLS Main

MAIN CALLS Display

DISPLAY CALLS validURL

VALIDURL CALLS each validation piece

Returns to display, returns to main

Checking order: (at each step check for invalid symbols; input reference value for output, return non 0 if error)

Check for protocol

Check for "://"

If fails, exit check

Check from valid list

If fails, continue to domain

Check for domain

Find domain beginning ("\n") and either port beginning (":") or filepath beginning ("/")

If fails, exit check

Split string to check format

If fails, evaluate next item

Check host from valid list

If fails, evaluate next item

Check for port

Check if exists beginning with (":") and ending with ("/")

If not included, check html or https

If format is bad, exit (cannot find filepath beginning)

Check number between 1 and 65535

If fails, evaluate next item

Check for filepath

Check beginning ("/") and end (".html" or ".htm")

If fails, exit (cannot validate filepath and optional parameter format)

If only one character, warn non existance of .htm(l), do not create error

Check for parameters

Check for formatting beginning with "?" and ending with alphanumeric value

Split into params

If fails, return multiple parameter error

Notes:

Get url, check, then (return 0 and append index) or (return 1 and append error and append index)