Addis Ababa Institute of Technology
School of Information Technology and Engineering
Fundamentals of Data Structures and Algorithms

Mini Assignment #1
(HTML Validator)

Submission Deadline 14/07/2022 Lunch Time (6:30 Local Time)

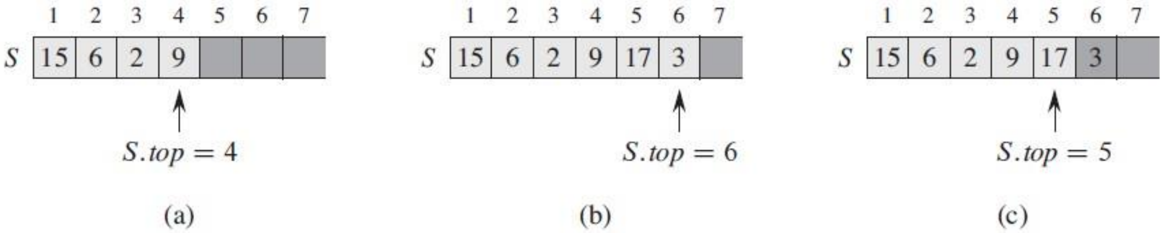
This activity introduces you to the applications of stack data structure, which has been covered in the lectures, to a real application.

Intro

A stack is a data structure that stores a linear collection of items such that the last item inserted is the last item removed: the stack implements a last-in, first-out, or LIFO, policy. Adding and removing items is restricted to one end known as the top of the stack.

The INSERT operation on a stack is often called PUSH, and the DELETE operation, which does not take an element argument, is often called POP. These names are allusions to physical stacks, such as the spring loaded stacks of plates used in cafeterias. The order in which plates are popped from the stack is the reverse of the order in which they were pushed onto the stack, since only the top plate is accessible.

As the figure below shows, we can implement a stack of at most n elements with a list. The list has an attribute S.*top* that indexes the most recently inserted element.



Task 1

Implement the stack data structure based on the starter code (‘stack.py’) provided.

- Stack(): Create a new empty stack
- Length(): Returns the number of items in the stack
- isEmpty(): Returns a Boolean value indicating if the stack is empty
- s_push(item): Adds the given item to the top of the stack
- s_pop(): Removes and returns the top item of the stack, If stack is not empty

- s_peek(): Returns ta reference to the item on top of the non-empty stack without removing it.

Task 2

HTML(Hyper Text Markup Language) is the standard markup language for documents designed to be displayed in a web browser.

XML was designed to store and transport data. XML is often used for distributing data over the internet. The standard constituents are tags.

Tags: markup either begin with the character < and end with a > and comes with three flavors

- start-tag: <section>
- end-tag: </section>
- empty-element <section/>

Element: an element is a logical document component that either begins with a start-tag and ends with a matching end-tag or only an empty-element tag.

```
<p>

  <strong>Bob</ strong >

    <em>Eve</em>

</p>
```

In this problem, you are required to validate the structure of an HTML file. In this task you are required to check if a start-tag is closed with an end-tag. You'll be using the stack data structure you implemented in the first task.

CLUES

While reading the HTML document if current read character is

1. '<' Character and the next one is different from '/' store it till you find '>' character (i.e start-tag) and push it to the stack.
2. '<' Character and the next one is '/' store it till you find '>' character (i.e end-tag). Check if the last start-tag is the same as the end-tag (the top element of the stack). If they are different stop execution and return line number and the invalid tags.
3. Some HTML tags contain inner attributes. Only the tag should be added and your code must be able to ignore the attributes.

   Example: <a href=”www.google.com” id=”link” > Google<a/>

   href and **id** are attributes of the tag a. The attributes must be ignored and only a must be pushed into the stack
4. If the stack is empty and the read finishes then the HTML is well structured.

The program you are going to write should be able to check if all of the tags in an HTML file are closed.

For example:

```
<html>
<p>Hello</p>
</html>
```

is a valid html. Because all of the tags are closed.

```
<html>
<p id="paragraph">Hello</p>
<img>Image</img>
</html>
```

This one is also a valid html tag

```
<html>
<p id="paragraph">
 <a href=”www.google.com”> Google<a/>
<div>
   <p>
         This is another content
   </p>
</div>
</p>
</html>
```

This one is also a valid HTML.

AN INVALID HTML file is the one that doesn't have tags that are closed.
If an opened tag is not closed or if a closing tag precedes an opening tag the HTML file is INVALID.

BONUS

Handle empty-elements

THE DEADLINE WILL NOT BE EXTENDED!
NO THIRD PARTY OR COPIED CODE IS ALLOWED
IF YOU CHEATED YOUR WORK WILL BE DISQUALIFIED

SUBMISSION INSTRUCTIONS

1. The file name should be section_number_yourname_lastname_ugr_id_12.py and your code on the docs file.
2. You are not allowed to use map functions.
3. Any late submission will result in disqualification.
4. Your code should be readable and the variable names should be more than three characters.

- **Any use of i,j,k ... x,y,z or any single letter as a variable is not allowed. (Even in loops)**
- **Variables should be clear.**

5. Your code must read from HTML file, not direct string input.

Submission Deadline 14/07/2022 Lunch Time (6:30 Local Time)

References

CLRS. (n.d.). *Introduction to Algorithms(CLRS) third edition.*

D.Necaise, R. (2010). *Data structures and Algorithms using python.*

W3C

Wikipedia