UI CHALLENGE

Using a Selenium-based implementation, build an automated UI test to order “Today’s Deals” on Amazon.com from lowest price to highest price.

Code to place order in Amazon

```
package com.amazon.pageobjects.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

import com.amazon.pageobjects.selection.AmazoniPhone6SearchResults;

public class AmazonHomePage extends AbstractPage {

private WebDriver driver;

public static final String URL = "https://www.amazon.co.uk/";

// Use annotations to locate elements.
@FindBy(id = "twotabsearchtextbox")
private WebElement twotabsearchtextbox;

@FindBy(css = "input.nav-submit-input")
private WebElement searchButton;

/**
* Sole Constructor
* @param driver
* the web driver
*/
public AmazonHomePage(WebDriver driver) {
super(driver);
this.driver = driver;
}

/**
* Returns the page URL.
*
* @return String representation of the URL.
*/
@Override
public String getPageURL() {
return URL;
}
```

```
/**
* Service: Perform search.
*
* @param text
* the search term.
* @return the search result page object.
*/
public AmazoniPhone6SearchResults searchFor(String text) {
twotabsearchtextbox.sendKeys(text);
twotabsearchtextbox.submit();
searchButton.click();
return new AmazoniPhone6SearchResults(driver);
}
```

Verify that products are sorted correctly in a browser

```
//create an LinkedList instead of arraylist because it preserves insertion order

List<WebElement> products_Webelement = new LinkedList<WebElement>();

//store the products (web elements) into the linkedlist

products_Webelement = d1.findElements(By.xpath("//img[contains(@id, 'product-collection-image')]"));

//create another linked list of type string to store image title

LinkedList<String> product_names =  new LinkedList<String>();

//loop through all the elements of the product_webelement list get it title and store it into the product_names list

for(int i=0;i<products_Webelement.size();i++){

String s = products_Webelement.get(i).getAttribute("alt");

product_names.add(s); }

//send the list to chkalphabetical_order method to check if the elements in the list are in alphabetical order

boolean result = chkalphabetical_order(product_names);

//print the result
System.out.println(result);
```

Verify that products have an image

```
@Test
public void CheckImage() throws Exception {
driver.get(baseUrl);
WebElement ImageFile = driver.findElement(By.xpath("//img[contains(@id,'Test Image')]"));
```

```
Boolean ImagePresent = (Boolean) ((JavascriptExecutor)driver).executeScript("return
arguments[0].complete && typeof arguments[0].naturalWidth != \"undefined\" &&
arguments[0].naturalWidth > 0", ImageFile);
if (!ImagePresent)
      {
          System.out.println("Image not displayed.");
      }
      else
      {
         System.out.println("Image displayed.");
      }
}
```

Verify that products have a price, and that its formatted properly

Price listed or not verification

```
List<WebElement> allItems=driver.findElements("xpath or id");

for (int i=0;i<allItems.size();i++)
{
If (allitems.get(i).isdisplayed)
{
Sos(" price is listed");
}
Else
Sos("price not available");
System.out.println(allItems.get(i).getText);
}
```