Testing und Debugging - Übungsblatt 2:

Gruppe 11:
Manuel Neubauer - 79358
Hueseyin Selcuk - 79375
Abdullah Yildiz - 79669
Daniel Zahl - 79356

1. **Definitions- und Wertebereich**
   1. Wieviele Werte für digit_string benötigt man mindestens um die Funktion vollständig zu testen? Begründen Sie Ihre Antwort.

- Es sind insgesamt 2 Werte notwendig um die Funktion vollständig zu Testen:
  Jeweils ein String mit gerader und ungerader Länge. Es müssen zahlen kleiner sowie größer-gleich 5 enthalten sein.

  Für l % 2 == 0: Für die erste If-Abfrage muss l (Länge) gerade sein:
  Danach folgt eine Verzweigung für l >= 1 in der for-Schleife:
  - Für ((i + 1) % 2== 0) ((i + 1) % 2 ist gerade) wird das if getriggert.
  - Andernfalls wird „luhn_digit“ aufgerufen. Hier bietet es sich an im Testwert eine Zahl >= 5 und eine Zahl < 5 zu haben, um beide Verzweigungen von „luhn_digit“ zu treffen.

  Für das else: Für das else kommen dann nur Zahlen mit ungerader l (Länge) in Frage:
  Danach folgt eine Verzweigung für l >= 1 in der for-Schleife:
  - Falls ((i + 1) % 2 == 0) ((i + 1) % 2 ist gerade) ist, wird „luhn_digit“ getriggert. Hier bietet es sich an im Testwert eine Zahl >= 5 und eine Zahl < 5 zu haben.
  - Andernfalls wird die else-Abfrage getriggert ((i + 1) % 2 ist ungerade).

2. Geben Sie die Werte für digit_string aus der vorgegangen Aufgabe explizit an?

- Wert 1: „0127“
- Wert 2: „01236“

3. Muss man beim Testen von luhm_checksum mit versteckten Eingaben rechnen? Wenn ja, wie muss dann exemplarisch ein Testfall aussehen, der dem Rechnung trägt? Wenn nein, begründen Sie kurz diese Antwort.

- Nein, es ist keine weitere Überprüfung notwendig. Der komplette Code liegt vor, und es sind keine unerwünschten Funktionen vorhanden.
  Es gibt weder zeitversetzte usereingaben noch andere zeitabhängige Funktionen/Funktionsaufrufe.
  Eine Interaktion mit Hardware oder Systemcalls mit dem Betriebssystem werden auch nicht verwendet. Somit ist nicht mit versteckten eingaben zu rechnen.

2. **GUI (Graphical User Interface) Testing**

1. Aus welchen Elementen besteht der Definitionsbereich beim Testen einer GUI Applikation

- Werte Eingaben durch User (Tastatur)
  z.B. (Textfelder, Tastenkombinationen, Definierte Hotkeys)
- Mausklicks (Buttons) durch User

2. Aus welchen Elementen besteht der Wertebereich?

- Fehlermeldungen, Bestätigungsmeldungen („Archiv entpackt“, „Vorgang abgeschlossen“), Bildschirmausgaben

3. Eine mögliche Sammlung von Tests für eine GUI Applikation wäre es, die Applikation gemäß ihrer Spezifikation, für 5 Minuten zu Benutzen (z.B. OpenOffice testen, können Sie einfach einen kurzen Brief schreiben und diesen dann formatieren). Nennen Sie zwei weitere Möglichkeiten für Sammlungen von Tests, um eine GUI Applikation zu testen. (je 5 Punkte)

- Langzeit Test: Verschiedene Personen testen die Software gemäß ihren Spezifikationen über mehrere Tage hinweg und protokollieren, ob Fehler auftreten.
- Testen außerhalb des Spezifikationsmaßes: Testen von nicht erwarteten Eingaben/ Userverhalten. (z.B. Erstellen von mehreren tausend Ebenen in Photoshop, Öffnen von nicht-Bildformaten, wie .wmv, .mp4, .docx etc.)

4. Muss man beim Testen von GUI Applikationen mit versteckten Eingaben rechnen? Wenn ja, wie muss dann exemplarisch ein Testfall aussehen, der dem Rechnung trägt? Wenn nein, begründen Sie kurz Ihre Antwort.

- Ja beim benutzen einer GUI kann kein Quellcode eingesehen werden. Es könnten Zeitversetzte User-Eingaben ankommen/übernommen werden. Dadurch können Eingaben entstehen, die nicht wie geplant funktionieren, weil z.B. die Funktion nicht dokumentiert oder nicht ausgiebig getestet wurde.

  Testfälle:
  Schnell aufeinanderfolgende Mausklicks, aufeinanderfolgende Tastenkombinationen, lange Eingabepausen

5. Wie gründlich müssen Graphical User Interfaces im Allgemeinen gestestet werden? Begründen Sie Ihre Antwort kurz

- Zum Testen sollte sich viel Zeit genommen werden. Fehler, die im Nachhinein entdeckt werden, kosten viel mehr Geld zum fixen als in der Testphase!
  Im Prinzip müssen alle möglichen Kombinationen der „täglichen Arbeit“ und mögliche Sonderfälle getestet werden.
  Die Kommunikation zwischen Frontend(GUI) und Backend findet meist in verschiedenen Threads statt. Dadurch ist mit versteckten Eingaben aufgrund zeitlicher Abhängigkeit zu rechnen. Insbesondere, da es sich zwischen GUI (Userland) und Betriebssystem/Kernel um eine Vertrauensgrenze handelt.