**Part -1: Identifying the problems**

The two small, local computing problems which I have identified are stated below:

**Problem – 1: PDF merger**

We frequently work with several PDF files in our everyday chores that need to be combined into one document for convenience of viewing, sharing, or printing. For example, it's not uncommon to combine bills, reports, or other documentation that are available as separate PDF files. It can be time-consuming to manually merge these files, particularly if there are a lot of them or if the procedure needs to be done frequently.

A Python script that enables users to choose numerous PDF files and combine them into one might be a solution to this issue. This might automate the PDF merging process and greatly increase workflow productivity.

**Why This Is Beneficial**:  
a) Timesaving: Manually merging PDFs, particularly when there are numerous files, can take a long time. Users can save time by automating the procedure with a script.  
b) Ease of Use: For people who might not feel comfortable utilizing more complex software solutions to merge PDFs, having a straightforward, user-friendly script can be beneficial.  
c) Useful Application: In business settings where regular document consolidation is necessary, this could be advantageous in addition to personal use.

**Libraries expected to use**: PyPDF2 for manipulating PDFs and tkinter for building a basic graphical user interface (GUI) that lets users choose files are two libraries that are anticipated to be used.

**Problem – 2: Keyboard shortcut customizer**

Many people use the same set of programs or commands to complete tasks on their computers on a regular basis. But not all programs offer keyboard shortcut customization, which can be annoying, particularly for power users who want a more efficient workflow. For instance, it could take multiple clicks or menu navigation to open a regularly used folder or carry out a repetitious task.  
  
Users could save time and improve the efficiency of their everyday computer chores by using a Python script that lets them design and modify keyboard shortcuts for their preferred programs or actions.

**Why This Is Beneficial**:  
a) Efficiency: System interactions would be considerably quicker if custom shortcuts could be assigned for regularly performed operations.  
b) Customization: By allowing users to tailor their workflow to their own requirements and habits, the system became more user-friendly.

**The libraries are anticipated to be used:**  tkinter for creating a straightforward user interface for the tool, and keyboard for recording and allocating shortcuts.

**Part -3: Reflection on PDF merger solution:**  
My choice for this assignment was to figure out how to combine several PDF files into one document. To develop a practical and effective solution, I had to learn how to handle external libraries and learn how to manipulate files in Python, which made this assignment suitably difficult.

It was a moderately difficult problem, but I could handle it. Because I had to learn how to handle files, control user input, and use other libraries, it wasn't very simple. It wasn't too difficult, though, because I was able to divide the challenge into small parts and it had a clear goal. The primary tasks included merging the PDFs using PyPDF2 and using Tkinter to create a user interface for simpler file selection.

The hardest part of this task was figuring out how to use the PyPDF2 library efficiently. I had trouble at first figuring out how to handle file inputs correctly and produce the merged PDF. Careful error handling was also necessary in a few cases when I had to deal with erroneous inputs or stop the user from choosing non-PDF files.

I used the program with a range of PDF files to test the finished product and make sure it could properly merge them. Additionally, I examined edge scenarios like choosing no files at all or trying to merge non-PDF files.

I had a great time working on this assignment. Being able to develop a completely functional application that addressed a real-world issue made me feel accomplished. In addition to improving my Python programming abilities, the experience helped me better grasp how to divide a project into smaller, more manageable jobs.

**References:**

* PyPDF2 Documentation: <https://pypi.org/project/PyPDF2/>
* Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>