Inhalt

[**Probleme** 2](#_Toc182081310)

[**Aufgaben und ihre Schwierigkeit** 2](#_Toc182081311)

[**Lösung & Konzepte** 2](#_Toc182081312)

[**Umgang mit Konflikten** 3](#_Toc182081313)

[**Lösung & Konzepte** 3](#_Toc182081314)

[**Richtige Schlüssel finden anhand der Werte in einem Dictionary** 3](#_Toc182081315)

[**Lösung & Konzepte** 4](#_Toc182081316)

[**Allgemeine Beschreibung des Programms** 4](#_Toc182081317)

[**Funktionsweise im Überblick** 4](#_Toc182081318)

[**1. Dateieingabe und Datenaufbereitung** 4](#_Toc182081319)

[**2. Initialisierung der Schwierigkeitswerte** 5](#_Toc182081320)

[**3. Korrektur der Schwierigkeitsränge** 5](#_Toc182081321)

[**4. Erstellung einer sortierten Klausur** 5](#_Toc182081322)

[**5. Ausgabe des Ergebnisses** 5](#_Toc182081323)

**Probleme**

**Aufgaben und ihre Schwierigkeit**Die Aufgaben in den Klausuren sind durch einfache Buchstaben des Alphabets dargestellt (z. B.: A, B, C usw.). Die Schwierigkeit der Aufgaben wird durch eine Reihenfolge  
gekennzeichnet, wie etwa „A < B < C“, was bedeutet, dass Aufgabe B schwieriger ist als A, und C schwieriger als B. In der Datei können solche Reihenfolgen beliebig lang sein, und es gibt mindestens eine dieser Ketten in jeder Datei und jede Kette muss aus mindestens 2 Aufgaben bestehen. Diese Darstellung erschwert jedoch den direkten Vergleich der Aufgaben, da sie lediglich relative Schwierigkeitsgrade angibt und nicht sofort als numerische Werte interpretierbar ist.

Um die Aufgaben in einer sinnvollen Reihenfolge nach Schwierigkeit anzuordnen, muss das Programm die Reihung der Aufgaben in numerische Schwierigkeitsstufen überführen. Dabei wird jeder Buchstabe als „Schwierigkeitswert“ interpretiert, der durch eine Analyse aller Klausuren in der Datei bestimmt wird. Nur so ist eine genaue und konsistente Einordnung der Aufgaben nach Schwierigkeit möglich.

**Lösung & Konzepte**  
Die Schwierigkeitswerte der Aufgaben werden in einem Dictionary im Format „Aufgabe: Schwierigkeitswert“ gespeichert. Diese Struktur eignet sich besonders gut, da jeder Schlüssel (also jede Aufgabe) eindeutig ist und nicht mehrfach vorkommen kann, was klare Zuordnungen ermöglicht. Außerdem ist das Dictionary „mutable“, das heißt, die Schwierigkeitseinträge können jederzeit verändert und aktualisiert werden. So lassen sich die Schwierigkeitsgrade der Aufgaben flexibel anpassen, was eine reibungslose und effiziente Verwaltung der Aufgabenreihenfolge ermöglicht.

Um das Dictionary Schwierigkeiten zu erstellen, wird jede Klausur einmal durchlaufen und die Klausuren werden anhand des Symbols „<“ in die einzelnen Aufgaben aufgeteilt. In einem for-Loop werden diese Aufgaben einzeln geprüft und als Schlüssel im Dictionary Schwierigkeiten angelegt, falls sie noch nicht vorhanden sind. Jede neue Aufgabe erhält dabei den Standard-Schwierigkeitswert 1.

Um das Schwierigkeitsranking der Aufgaben korrekt festzulegen, wird jede Klausur einmal durchlaufen und die Schwierigkeitswerte der einzelnen Aufgaben angepasst. Die Aufgaben in jeder Klausurzeile werden durch das Zeichen „<“ getrennt, dass eine aufsteigende Schwierigkeit anzeigt. Im for-Loop werden die Aufgaben der Reihe nach überprüft und verglichen, um sicherzustellen, dass der Schwierigkeitswert der rechten Aufgabe mindestens um eins höher ist als der der linken.

#### **Für jede Klausurzeile werden folgende Schritte ausgeführt:**

##### **Aufgabenteilung und -länge:**

Die Aufgaben innerhalb der Zeile werden in eine Liste aufgeteilt, sodass sie paarweise verglichen werden können.

**Vergleich der Schwierigkeitsstufen:**

Für jede Aufgabe wird der Schwierigkeitswert (schwierigkeit\_links) mit dem der nächsten (schwierigkeit\_rechts) verglichen.

##### **Anpassung des Schwierigkeitswertes:**

Wenn die Schwierigkeit der linken Aufgabe kleiner oder gleich der rechten ist, wird der Schwierigkeitswert der linken Aufgabe erhöht, sodass er mindestens um 1 größer ist als der rechten. Dies stellt sicher, dass die Reihenfolge der Schwierigkeiten in Übereinstimmung mit den relativen Anforderungen bleibt.

Durch diesen einmaligen Durchlauf werden die Schwierigkeitswerte an die festgelegte Reihenfolge angepasst, was eine valide und logische Schwierigkeitseinstufung der Aufgaben sicherstellt.

## **Umgang mit Konflikten**

In der vorliegenden Aufgabe sollen Klausuren basierend auf ihren Schwierigkeitsstufen sortiert werden. Dabei treten Konflikte auf, wenn verschiedene Klausuren unterschiedliche Anforderungen an den Schwierigkeitsgrad einer bestimmten Aufgabe stellen.

Ein Konflikt entsteht, wenn beispielsweise eine Klausur festlegt, dass A < B gilt, während eine andere Klausur B < A vorgibt, wodurch sich die Schwierigkeit von B im Vergleich zu anderen Aufgaben verändert. Da jede Klausur nur relative Schwierigkeitsverhältnisse innerhalb ihrer eigenen Reihenfolge angibt, können widersprüchliche Anordnungen auftreten, wenn Aufgaben in mehreren Klausuren in unterschiedlichen Konstellationen vorkommen.

Das Problem besteht also darin, eine konsistente Schwierigkeitsliste zu erstellen, obwohl die Klausuren teilweise widersprüchliche Schwierigkeitsanforderungen enthalten.

**Lösung & Konzepte**

Das Problem der widersprüchlichen Schwierigkeitsanordnungen wird im Programm automatisch gelöst, indem die Klausuren nacheinander durchlaufen werden und die zuletzt gelesene Klausur als maßgeblich betrachtet wird. Dadurch wird die Schwierigkeitseinstufung jeder Aufgabe immer wieder angepasst, wenn eine neue Klausur eine andere relative Reihenfolge vorgibt.

Jede Klausur aktualisiert also das Schwierigkeitsranking der Aufgaben, indem die Schwierigkeit jeder nachfolgenden Aufgabe im Vergleich zur vorherigen um mindestens eins erhöht wird. Wenn eine Klausur eine bestehende Schwierigkeit überschreiben muss, wird diese Anpassung direkt vorgenommen, sodass das Dictionary schwierigkeiten immer die aktuelle, konfliktfreie Rangfolge enthält. Am Ende ergibt sich ein konsistentes Ranking, in dem die Schwierigkeitsverhältnisse der zuletzt betrachteten Klausur übernommen wurden. Dieses einfache Verfahren verhindert auf effiziente Weise Konflikte, ohne dass komplexe Abgleichs Mechanismen notwendig sind.

**Richtige Schlüssel finden anhand der Werte in einem Dictionary**

Ein zentrales Problem bei der Implementierung des Programms ist die Notwendigkeit, die richtigen Schlüssel aus einem Dictionary zu finden, basierend auf den zugehörigen Werten. Das Dictionary repräsentiert die Schwierigkeitsgrade von Aufgaben, wobei jeder Schlüssel (Aufgabe) einem spezifischen Schwierigkeitswert zugeordnet ist. Die Herausforderung besteht darin, die Schlüssel zu identifizieren, die einer bestimmten Zielschwierigkeit entsprechen, und gleichzeitig mehrere Bedingungen zu beachten.

**Lösung & Konzepte**

#### **Die Funktion finde\_schluessel\_durch\_schwierigkeit ist dafür zuständig, einen Schlüssel zu finden, der den folgenden Anforderungen entspricht:**

##### **Übereinstimmung mit der Zielschwierigkeit:**

Der Schlüssel muss den Schwierigkeitswert aufweisen, der in ziel\_schwierigkeit angegeben ist. Dies stellt sicher, dass nur die Aufgaben mit dem gewünschten Schwierigkeitsgrad in Betracht gezogen werden.

##### **Enthalten in den zuzuordnenden Aufgaben:**

Der gefundene Schlüssel muss in der Liste anzuordnende\_aufgaben enthalten sein. Dies ist wichtig, um sicherzustellen, dass nur die Aufgaben, die tatsächlich angeordnet werden sollen, berücksichtigt werden.

##### **Nicht bereits verwendet:**

Der Schlüssel darf nicht in der Liste benutzte\_schluessel enthalten sein. Diese Bedingung sorgt dafür, dass jede Aufgabe nur einmal in der Endausgabe verwendet wird, um Duplikate zu vermeiden.

Wenn alle Bedingungen erfüllt sind, wird der gefundene Schlüssel der Liste der verwendeten Schlüssel hinzugefügt, und der Schlüssel wird zurückgegeben. Andernfalls wird None zurückgegeben, falls kein passender Schlüssel gefunden werden kann.

Dieses Verfahren sorgt dafür, dass die Aufgaben zuverlässig anhand ihrer Schwierigkeit identifiziert werden können, da hierfür Zahlenwerte zur Unterscheidung benutzt werden. Eine fehlerhafte Zuordnung dieser Schlüssel-Wert-Paare könnte jedoch zu einer ungültigen Zusammenstellung der Klausur führen.

**Allgemeine Beschreibung des Programms**

Das Programm analysiert und sortiert Aufgaben einer Klausur nach ihrem Schwierigkeitsgrad basierend auf festgelegten Beziehungen, die in einer Textdatei definiert sind. Die Hauptfunktion des Programms besteht darin, eine Schwierigkeitsreihenfolge für Aufgaben zu erstellen und eine valide Klausur auszugeben, die den relativen Schwierigkeitsgraden entspricht.

**Funktionsweise im Überblick**

**1. Dateieingabe und Datenaufbereitung**

Das Programm liest eine Textdatei ein und speichert alle Zeilen in einer Liste zeilen.

Nach dem Entfernen von Leerzeichen und Zeilenumbrüchen werden die Werte n, m, und k aus der ersten Zeile gelesen (wobei nur k relevant ist).

Die letzte Zeile enthält eine Liste von Aufgaben, die sortiert werden sollen, und wird separat als anzuordnende\_aufgaben gespeichert.

Anschließend werden die erste und die letzte Zeile aus zeilen entfernt, sodass nur noch die Schwierigkeitsbedingungen verbleiben.

**2. Initialisierung der Schwierigkeitswerte**

Ein Dictionary schwierigkeiten wird erstellt, um jeder Aufgabe einen Schwierigkeitswert zuzuweisen. Jede Aufgabe beginnt standardmäßig mit dem Schwierigkeitswert 1.

Der Code durchläuft jede Klausurzeile und fügt jede Aufgabe in das Dictionary ein, falls sie noch nicht vorhanden ist, um sicherzustellen, dass alle Aufgaben initial erfasst sind.

**3. Korrektur der Schwierigkeitsränge**

Ein weiterer Schleifendurchlauf durch die Klausuren überprüft und korrigiert die Schwierigkeitswerte. Für jede Zeile wird geprüft, ob die Schwierigkeitsreihenfolge stimmt (< bedeutet, dass die rechte Aufgabe schwieriger sein sollte als die linke).

Falls die Schwierigkeit nicht den Anforderungen entspricht, wird der Schwierigkeitswert der rechten Aufgabe erhöht, sodass die Reihenfolge stimmt.

**4. Erstellung einer sortierten Klausur**

Eine Liste ranking wird erstellt, die die Schwierigkeitswerte der zu sortierenden Aufgaben (laut anzuordnende\_aufgaben) enthält und diese Werte sortiert.

Eine Funktion finde\_schluessel\_durch\_schwierigkeit sucht basierend auf dem Schwierigkeitswert den zugehörigen Aufgabenschlüssel, überprüft, ob die Aufgabe bereits benutzt wurde, und fügt sie dann der gültigen Klausur hinzu.

**5. Ausgabe des Ergebnisses**

Zum Abschluss gibt das Programm das finale Schwierigkeitsranking aller Aufgaben und die valide Klausur als String aus, die die Aufgaben in der Schwierigkeitsreihenfolge einfach zu schwer enthält.