



Klausur „Programmiersprachen 1“

Praktischer Teil

Dauer: 60 Minuten

Erlaubte Hilfsmittel: Nicht limitiert, insbesondere auch das verfügbare Vorlesungsmaterial der Veranstaltung unter <http://www.fh-wedel.de/mitarbeiter/mri/prog-1/> und Ihre eigenen Lösungen aus dem Unterricht.

Lösen Sie die Aufgabe auf ihrem Rechner mit der **Delphi XE2-Entwicklungsumgebung**. Laden Sie nach der Bearbeitung die **.dpr-Datei auf den Server**. Prüfen Sie anschließend durch einen erneuten Download, ob Sie die korrekten Quellen hochgeladen haben.

Es sind in diesem Teil insgesamt maximal **17 Punkte** zu erreichen. Bei jeder Aufgabe ist die dort **mögliche Punktzahl** angegeben, die somit auch eine ungefähre Aufwandsabschätzung ermöglicht. Sollte eine **Lösung deutlich umfangreicher** sein, als die vorhandene Maximalpunktzahl annehmen lässt, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Ihnen **Aufgabenstellungen missverständlich** erscheinen, so legen Sie bitte Ihre Interpretation der Aufgabe kurz schriftlich dar und lösen Sie die Aufgabe entsprechend.



1. Teilmengen und vollkommene Zahlen

Schreiben Sie ein Programm zur Berechnung von Teilmengen. Die Reihenfolge in der die Teilaufgaben gestellt werden ist ein expliziter Hinweis auf eine sinnvolle Bearbeitungsreihenfolge. Sofern Sie bei einer Aufgabe keinen geeigneten Ansatz finden, können Sie diese bei einigen Aufgaben auch überspringen. In diesem Fall erhalten Sie für die konkrete Teilaufgabe keine Punkte, können aber die nachfolgenden Aufgabenteile bearbeiten.

Um Missverständnisse zu vermeiden:

- Legen Sie in Ihrem Programmkopf als Kommentar dar, welche Teilaufgaben Sie bearbeitet haben.
- Schreiben Sie Ihren Namen und die Matrikelnummer als Kommentar in den Programmkopf.

Allgemeiner Hinweis: Sie können davon ausgehen, dass der Benutzer keine Eingaben außerhalb des 32-Bit-Wertebereichs vornimmt.

a) Benutzereingabe einer Zahl

3 Punkte

Fragen Sie vom Benutzer eine ganze Zahl > 10 ab und weisen Sie explizit auf die Grenze hin. Sofern eine negative Zahl eingelesen wird, soll der Benutzer erneut um eine Eingabe gebeten werden. Eingabefehler aufgrund falscher Datentypen (Buchstaben, reelle Zahlen) müssen Sie für diese Teilaufgabe nicht behandeln. Speichern Sie den erfragten Wert in einer Variable `teilerMin`.

Sollten Sie diesen Teil nicht lösen können: Deklarieren Sie eine Konstante `BENUTZEREINGABE` und weisen Sie dieser den Wert 253 zu. Weisen Sie dann der Variable `teilerMin` den Wert `BENUTZEREINGABE` zu.

b) Absturzfremde Benutzereingabe einer Zahl

2 Punkte

Lesen Sie den Wert des Benutzers zunächst in einen `string` ein und überführen Sie den Wert anschließend in die Variable `teilerMin`. Sofern der Benutzer nicht erlaubte Zeichen eingibt, ist er auf das erste inkorrekte Zeichen hinzuweisen.

Beispiel: Im Erfolgsfall ist die Ausgabe unverändert. Wenn der Benutzer jedoch zum Beispiel „12A“ eingibt, soll die Ausgabe ihn auf ein fehlerhaftes Zeichen an der dritten Stelle hinweisen.

Sollten Sie diesen Teil nicht lösen können: Überspringen Sie diesen Aufgabenteil. Sie können ihn auch später noch sehr gut isoliert bearbeiten, sofern Ihnen die Zeit dafür bleibt.



c) Teiler einer Zahl

3 Punkte

Geben Sie alle Teiler der in `teilerMin` hinterlegten Zahl in der folgenden Schreibweise aus:

`T_<teilerMin> = { <Komma-separierte, aufsteigende Teiler von teilerMin> }`

Eine Zahl wird als Teiler einer anderen Zahl bezeichnet, wenn sie diese ganzzahlig ohne Rest teilt. Beachten Sie, dass nach dem letzten Teiler kein Komma ausgegeben werden soll.

Beispiel: 5 ist kein Teiler von 253, da 253 „durch“ 5 = 50 Rest 3. 11 ist hingegen ein Teiler von 253, da 253 „durch“ 11 = 23 Rest 0. Die vollständige Teilermenge von 253 in der korrekten Schreibweise lautet:

`T_253 = {1, 11, 23, 253}`

d) Optimierung der Generierung einzelner Teilermengen

1 Punkte

Trivialerweise sind für eine beliebige Zahl n die Zahl 1 und n selbst Teiler. Nutzen Sie diesen Umstand um bei der Berechnung einer Teilermenge nicht mehr als $(n / 2) + 1$ Zahlen auf Ihre Teilereigenschaft zu prüfen.

Sollten Sie diesen Teil nicht lösen können: Überspringen Sie diesen Aufgabenteil.

e) Benutzereingabe eines Intervalls

2 Punkte

Erweitern Sie die Eingabe um die Abfrage einer weiteren Zahl, welche Sie in der Variablen `teilerMax` speichern. Diese Zahl muss unmittelbar nach der Eingabe ebenfalls > 10 sein. Sie muss jedoch nicht zwingend größer als die zuvor eingegebene Zahl `teilerMin` sein. Stattdessen sollen die nach der Eingabe verglichen und gegebenenfalls getauscht werden, so dass gilt `teilerMin <= teilerMax`. Geben Sie das Intervall in der Schreibweise `[<teilerMin>, <teilerMax>]` aus.

Beispiel: Der Benutzer gibt die Zahlen 12 und 21 ein, in diesem Fall müssen die Werte nicht getauscht werden, die Ausgabe lautet daher `[12, 21]`. Sollten hingegen die Zahlen 21 und 12 eingegeben werden ist eine Vertauschung notwendig, das sich ergebende Intervall ist jedoch identisch.

Sollten Sie diesen Teil nicht lösen können: Verzichten Sie auf das Einlesen eines zweiten Wertes und berechnen Sie `teilerMax := teilerMin + 100`, sofern Sie den nächsten Aufgabenteil lösen möchten.



f) Alle Teilmengen eines Intervalls

2 Punkte

Berechnen Sie die Teilmengen für alle Werte des vom Benutzer eingegebenen Intervalls

[teilerMin, teilerMax]. Geben sie dabei jede Teilermenge auf einer einzelnen Zeile aus.

Sollten Sie diesen Teil nicht lösen können: Überspringen Sie diesen Aufgabenteil.

g) Hinweis auf vollkommene Zahlen

2 Punkte

Eine Zahl ist vollkommen, wenn die Summe ihrer Teiler das Doppelte der Zahl selbst ergibt. Geben Sie nach der Berechnung der Teilermenge diese Summer der Teiler aus. Sofern es sich um eine vollkommene Zahl handelt soll zusätzlich der string „VOLLKOMMEN“ ausgegeben werden.

Beispiel: Die kleinste vollkommene Zahl ist 6, da ihre Teiler 1, 2, 3, 6 aufsummiert 12 ergeben. Die Zahl 25 hat hingegen die Teiler 1,5, 25. Diese ergeben in der Summe nicht 50, daher ist die 25 keine vollkommene Zahl. Die nächsten vollkommenen Zahlen sind 28 und 496.

Sollten Sie diesen Teil nicht lösen können: Überprüfen Sie andere übersprungene Teile. Sie können den folgenden Teil ohne diesen Teil nicht sinnvoll bearbeiten.

h) Unterdrücken von unvollkommenen Zahlen

2 Punkte

Hinweis: Für diesen Aufgabenteil müssen Sie Veränderungen an dem Ablauf der Ausgabe vornehmen. Die Ausgabe ist aber gleichzeitig ein elementares Bewertungskriterium. Machen Sie daher unbedingt vor der Bearbeitung eine Sicherheitskopie ihres aktuellen Arbeitsstandes.

In dieser Aufgabe wird die Ausgabe der Teilmengen für alle Zahlen unterdrückt, bei denen es sich nicht um vollkommene Zahlen handelt. Um eine redundante Berechnung der Teiler zu vermeiden, sollen die Ausgaben für die in einem String zwischengespeichert werden. Geben Sie diesen String nach Berechnung der Teilersumme nur aus, wenn es sich um eine vollkommene Zahl handelt. Sofern Sie die Teiler mehrfach berechnen, wird ein Punkt abgezogen.