

Assignment 7

Programmieren 1 – WiSe 25/26

Prof. Dr. Michael Rohs, Falk Stock, M.Sc.

Alle Assignments (bis auf das erste) müssen in Zweiergruppen bearbeitet werden. Ein Gruppenmitglied kann dabei die Lösung der Zweiergruppe gebündelt abgeben. Einzige Ausnahme ist dabei das erste Assignment, wo jedes Gruppenmitglied einzeln abgeben muss. Namen beider Gruppenmitglieder müssen sowohl in der PDF-Abgabe, als auch als Kommentar in jeglichen Quelltextabgaben genannt werden. Plagiate führen zum Ausschluss von der Veranstaltung.

Abgabe bis Donnerstag den 11.12. um 23:59 Uhr über <https://assignments.hci.uni-hannover.de/WiSe2025/Prog1>. Die Abgabe muss aus einer einzelnen ZIP-Datei bestehen, die den Quellcode, eine PDF für Freitextaufgaben und alle weiteren nötigen Dateien (z.B. Eingabedaten oder Makefiles) enthält. Lösen Sie Umlaute in Dateinamen auf.

Zum Bestehen der Studienleistung müssen Sie mindestens zwei von vier Punkten pro Assignment erzielen. Möchten Sie zusätzlich den Klausurbonus erreichen, müssen Sie über alle Assignments hinweg 75% der Punkte erreichen.

Aufgabe 1: Rechtecke

In dieser Aufgabe werden Rechtecke durch `structs` dargestellt und verarbeitet. Rechtecke besitzen zwei Koordinaten (`x` und `y`), eine Höhe (`h`) und eine Breite (`w`). Alle Werte werden als `int` repräsentiert. Orientieren Sie sich an den im Skript unter Recipe for Compound Data beschriebenen Schritten. Verwenden Sie die Template-Datei `rectangles.c` und bearbeiten Sie die mit `todo` markierten Stellen.

- Vervollständigen Sie die Definition der Struktur `Rectangle` und implementieren Sie die Funktion `make_rectangle`. Stellen Sie durch Preconditions sicher, dass sowohl X- und Y-Koordinate, als auch Breite und Höhe des Rechtecks nicht auf negative Werte gesetzt werden können.
- Implementieren Sie die Funktion `intersects`, in der geprüft wird, ob sich zwei Rechtecke überschneiden.
- Implementieren Sie die Funktion `print_rectangle`, in der Sie ein gegebenes Rechteck auf der Kommandozeile graphisch skizzieren. Die Breite und Höhe des Rechtecks werden als Anzahl der Zeichen auf der Horizontalen oder als Anzahl der Zeilen auf der Vertikalen interpretiert. Die X-Koordinate wird durch Leerzeichen am Beginn jeder Zeile und die Y-Koordinate durch leere Zeilen über dem Rechteck repräsentiert.
Über die Struktur `DrawOptions` wird die Darstellung konfiguriert. Rechtecke können mit oder ohne Rahmen gezeichnet werden. Der String `fill_pattern` bestimmt die Füllung des Rechtecks. Eine Beispielausgabe ist unten gegeben, weitere finden Sie in der Template-Datei.
- (OPTIONAL) Implementieren Sie die Funktion `make_draw_options_range`. Diese soll `DrawOptions` erzeugen, deren Pattern durch eine Range von Zeichen definiert ist. Lautet beispielsweise das

from-Zeichen "1" und das to-Zeichen "8", wird das Pattern "12345678" erzeugt. Lautet das from-Zeichen "e" und das to-Zeichen "k", wird das Pattern "efghijk" erzeugt. Stellen Sie über preconditions sicher, dass beide Zeichen in dem Intervall [32, 126] liegen und from < to gilt. Das Rechteck soll mit Rahmen gezeichnet werden.

Geben Sie mindestens 6 verschiedene Beispielrechtecke auf der Konsole aus. Stellen Sie sicher, dass jede DrawOptions-Variante mindestens einmal vorkommt.

Beispielausgabe der Funktion `print_rectangle`:

```
+-----+
|       |
|       |
|       |
+-----+
```

Das Rechteck wurde hier über den Aufruf `make_rectangle(0, 0, 9, 5)` erzeugt und das Pattern über `make_draw_options_fill(" ")`.

Aufgabe 2: Operationen auf Arrays (1 Punkt)

In dieser Aufgabe sollen Sie verschiedene Operationen für Integer-Arrays implementieren. Das Template für diese Aufgabe ist `array_operations.c`. Testfälle für die verschiedenen Aufgabenteile sind bereits vorhanden und können gegebenenfalls ergänzt werden.

- Implementieren Sie zuerst die Funktion `bool compare()`. Diese soll zwei Arrays `array_a` und `array_b` vergleichen und `true` zurückgeben, wenn der Inhalt beider Arrays identisch ist. Andernfalls soll `false` zurückgegeben werden. Diese Funktion ist relevant für die weiteren Teilaufgaben, da sie auch in den Testfällen der späteren Teilaufgaben verwendet wird. Bearbeiten Sie die weiteren Teilaufgaben erst, wenn Sie sicher sind, dass `bool compare()` funktioniert (Alle Testfälle werden erfüllt).
- Implementieren Sie die Funktion `int remove_negatives()`, die ein Array übergeben bekommt und in diesem in-place die negativen Zahlen entfernt. Dafür werden die verbleibenden positiven Zahlen an den Anfang des Arrays verschoben und die Anzahl an positiven Elementen als neue Länge zurückgegeben. Sie reservieren also keinen neuen Speicher, sondern verändern das bestehende Array.
- Machen Sie sich mit der Struktur `BetterArray` vertraut und implementieren Sie danach die Funktion `BetterArray intersect()`. Diese bekommt zwei Arrays übergeben und bildet die Schnittmenge aus beiden. Die Schnittmenge wird auch hier in-place in einem der übergebenen Arrays gespeichert. Wählen Sie dazu das Array aus, das eine kleinere Länge hat. Sollten beide Arrays gleichlang sein, so überschreiben Sie `array_a`. Geben Sie dann eine entsprechend initialisierte Struktur `BetterArray` mit dem Ergebnis und der Länge des Ergebnisses zurück.

- d) Implementieren Sie die Funktion `void merge_sorted_arrays()`, die die beiden sortierten Arrays `array_a` und `array_b` zu einem sortierten Array `result` zusammenfügt.
- e) Welche Vorteile bietet die Struktur `BetterArray`? Wie werden Strings in C repräsentiert? Wie ist in C die Länge eines Strings gespeichert bzw. wie wird sie bestimmt? Ist so etwas auch für `int []` möglich? Beantworten Sie diese Fragen als Kommentar in Ihrer Quelltextdatei.

Aufgabe 3: Einlesen und Verarbeiten von Messwerten

In dieser Aufgabe soll die Tabelle `measurements.txt` mit Messdaten aus einem Versuch eingelesen und verarbeitet werden. Gemessen wurde die benötigte Zeit, um eine Eingabe mit verschiedenen Eingabegeräten zu tätigen. Die Tabelle enthält eine Zeile mit Spaltenüberschriften gefolgt von einer größeren Menge an Zeilen mit entsprechenden Werten. Die erste Spalte enthält eine Altersangabe (Typ `int`). Die zweite kodiert, ob bei dem Versuchsdurchlauf eine Maus ("m"), ein Touchscreen ("t") oder die Tastatur ("k") eingesetzt wurde. Die letzte Spalte gibt die benötigte Zeit (Typ `double`) an.

Berechnet werden sollen die folgenden statistischen Angaben:

- Das durchschnittliche Alter
- Die Standardabweichung der Alter
- Die durchschnittliche Zeit mit der Maus
- Die durchschnittliche Zeit mit dem Touchscreen
- Die durchschnittliche Zeit mit der Tastatur

Verwenden Sie die Template-Datei `measurements.c` und bearbeiten Sie die mit `todo` markierten Stellen.

- a) Entwerfen Sie eine Struktur namens `Statistics`, die die genannten Angaben repräsentieren kann.
- b) Implementieren Sie eine Konstruktorfunktion `make_statistics`, welche die Komponenten der Struktur auf 0 bzw. 0.0 initialisiert.
- c) Implementieren Sie die Funktion `print_statistics`. Geben Sie jeden Wert des `structs` mit kurzer Beschreibung auf je einer Zeile aus. Geben Sie Fließkommazahlen mit maximal zwei Nachkommastellen aus.
- d) Implementieren Sie die Funktion `compute_statistics`, welche die in einer Zeichenkette (Typ `String`) vorliegende Tabelle verarbeitet, die Statistik berechnet und diese zurückgibt. Schauen Sie sich zunächst an, wie der Inhalt von `measurements.txt` formatiert ist. Ihre Funktion muss nur für dieses Format funktionieren. Eine Fehlerbehandlung muss nicht implementiert werden.

Hinweise:

- Benutzen Sie zur Verarbeitung des Strings die Funktionen `s_length`, `s_get` und `s_sub`.
- Benutzen Sie zur Konvertierung eines Strings in ein `int` die Funktion `i_of_s` und zur Konvertierung eines Strings in ein `double` die Funktion `d_of_s`.
- Quadratwurzeln werden mit der C Funktion `sqrt` berechnet.

Aufgabe 4: Verschlüsselung durch Substitution

Eine Methode zur Verschlüsselung von Texten ist die Substitution einzelner Zeichen (https://de.wikipedia.org/wiki/Monoalphabetische_Substitution). Der Text wird zeichenweise durchgegangen. Jedes Zeichen wird dabei mit einem anderen vorgegebenen Zeichen ersetzt. Die Substitutionsregel ist in dem Template bereits in dem konstanten String `permutation` gegeben. Die Position der Zeichen in dem String bestimmt den ursprünglichen Buchstaben von A bis Z. In diesem Beispiel wird also der Buchstabe A substituiert mit S, B mit G, Z mit C usw.

Verwenden Sie die Template-Datei `substitution.c`. Bearbeiten Sie die mit `todo` markierten Stellen.

- Implementieren Sie die Funktion `encrypt`. Diese soll die gegebene Substitution auf die Eingabe anwenden. Schreiben Sie ein Purpose Statement.
- Implementieren Sie die Hilfsfunktion `s_index_of`. Diese soll den Index des Strings `s` zurückgeben, an dem das Zeichen `c` das erste Mal auftritt. Wenn das Zeichen in dem String nicht vorkommt, soll `-1` zurückgegeben werden.
- Implementieren Sie die Funktion `decrypt`. Diese soll die Substitution rückgängig machen. Schreiben Sie ein Purpose Statement.

Hinweise:

- Es sollen ausschließlich Großbuchstaben behandelt werden. Satz- und Leerzeichen sollen erhalten bleiben.
- Nutzen Sie die Funktionen `s_get` und `s_set` zum zeichenweisen Zugriff auf Strings.
- Nutzen Sie die Funktion `s_copy` um eine schreibbare Kopie eines Strings zu erhalten.
- Sie können mit Werten vom Typ `char` wie mit anderen Zahlenwerten rechnen.
Z.B. gilt: `'A' + 1 == 'B'`