

Imperative Programmierung

–Aufgabenblatt 05 (44 + 1 Punkte)–

Hinweise

Die Lösungen der Aufgaben sind als PDF-Dokument bzw. C-Quelltext mit Hilfe des Versionskontrollsystems Subversion (SVN) abzugeben. Platzieren Sie das PDF-Dokument mit ihren Antworten im Ordner `a05` innerhalb Ihres Gruppenverzeichnisses¹. Platzieren Sie die C-Quelltexte im Unterordner `a05/src`. C-Quelltexte müssen fehlerfrei mit den Optionen `-pedantic -Wall -Werror -std=c99` kompiliert werden können. Schreiben Sie in die abgegebenen Dateien die Namen und Matrikelnummern aller Gruppenmitglieder. Verspätete Abgaben oder Abgaben ohne Matrikelnummer werden nicht gewertet! Plagiate jeglicher Art führen zur Meldung beim Prüfungsausschuss und zum Nichtbestehen des Moduls.

Ausgabe: 22.11.2024

Abgabe: 08.12.2024 bis 23:59 Uhr

Aufgabe 1 - Sortierte Arrays (6 Punkte)

Schreiben Sie einige Funktionen um Arrays aufsteigend sortierter, ganzer Zahlen zu verwalten. Beispielsweise ist das Array `{1, 4, 8}` aufsteigend sortiert; `{4, 1, 8}` hingegen nicht. Auch hier wird zusammen mit dem Array immer auch seine Länge separat an die Funktionen übergeben.

- (a) Implementieren Sie die Funktion `int is_sorted(int array[], int n)` die feststellt, ob ein übergebenes Array `array` der Länge `n` aufsteigend sortiert ist. (2 Punkte)
- (b) Implementieren Sie die Funktion `void print_array(int arra[], int n)`, die ein Array `array` der Länge `n` via `printf` in der Form `{1, 2, 3, 4}` ausgibt. Achten Sie auf die Kommasetzung und Leerzeichen in der Ausgabe. (2 Punkte)
- (c) Implementieren Sie eine Funktion `insert_sort`, der ein sortiertes Array `array`, seine Länge `n` und eine ganze Zahl `x` übergeben wird. Es soll ein sortiertes, um eine Stelle größeres Array erzeugt werden, das `x` sowie alle Elemente aus `a` enthält. Beispielsweise entsteht aus dem Array `{1, 4, 8}` der Länge 3 durch `insert_sort` mit dem Wert 5 das neue Array `{1, 4, 5, 8}` der Länge 4. Geben Sie dieses Array mittels der `print_array` Funktion aus. Zurückgeben muss ihre Funktion es aber nicht. (2 Punkte)
- (d) Bonus: Wie viele Elemente hat das größte Array von ganzen Zahlen, das Sie auf Ihrem Computer anlegen können? Begründen Sie, warum größere Arrays nicht möglich sind. (1 Bonuspunkt)

Aufgabe 2 - Anagramme (7 Punkte)

Ein Wort `W` ist ein Anagramm des Wortes `V`, wenn `W` aus `V` durch die Umstellung einzelner Buchstaben gebildet werden kann. Beispielsweise sind `Ampel`, `Lampe` und `Palme` Anagramme voneinander. Schreiben Sie ein Programm, um Anagramme zu erkennen. Zeichen, die keine Buchstaben sind, können Sie ignorieren. Es wird die ASCII-Codierung der Zeichen angenommen. Die Großbuchstaben haben die Dezimalkodierungen 65 ('A') bis 90 ('Z') und die Kleinbuchstaben 97('a') bis 122('z').

- (a) Implementieren Sie eine Funktion `letter_number`, die ein Zeichen übergeben bekommt und für einen Buchstaben in Groß- oder Kleinschreibung zurückgibt, um den wievielten Buchstaben des Alphabets es sich handelt. Dabei soll für 'a' und 'A' 0 zurückgegeben werden, für 'b' und 'B' 1 und so weiter. Falls es sich bei dem Zeichen nicht um einen Buchstaben handelt, soll `-1` zurückgegeben werden. (2 Punkte)

¹Ihr Gruppenverzeichnis ist unter <https://svn.informatik.uni-rostock.de/lehre/ip2024/groups/<group>/> mit `<group> ∈ {01, ..., 57}` zu finden.

- (b) Implementieren Sie eine Funktion `char_histogram`, die einen String und ein leeres Ganzzahlarray mit 26 Stellen übergeben bekommt. In dem Array soll die absolute Häufigkeit der Buchstaben des Strings gespeichert ist. Im Index 0 soll die Häufigkeit des Buchstabens 'A' bzw. 'a' stehen, in 1 die Anzahl von 'B's bzw. 'b's und so weiter. Zeichen, die keine Buchstaben sind, sollen ignoriert werden. (3 Punkte)
- (c) Implementieren Sie eine Funktion `is_anagram`, der zwei Worte als Strings und deren Länge übergeben werden und die überprüft, ob die Worte Anagramme voneinander sind. Verwenden Sie die Funktion `char_histogram` aus der vorigen Teilaufgabe. Testen Sie ihre Implementierung für das o.g. Beispiel und einem zusätzlichen Wort ihrer Wahl, welches kein Anagramm von Ampel ist. (2 Punkte)

Aufgabe 3 - Wir basteln uns ein Grafikprogramm (31 Punkte)

- (a) Schreiben Sie eine Funktion `int contained(double x, double a, double b)`, die eine 1 liefert, falls x in dem geschlossenen Intervall $[a, b]$ liegt, sonst 0. (Die Intervallgrenzen müssen nicht notwendigerweise in der richtigen Reihenfolge angegeben sein.) (3 Punkte)
- (b) Schreiben Sie eine Funktion `int line(double x, double y, double px, double py, double qx, double qy)`, die eine 1 liefert, falls der Abstand zwischen dem Punkt $X = (x, y)$ und der Strecke \overline{PQ} kleiner als 0.5 ist und sonst 0. (Hierbei gilt: $P = (px, py)$, $Q = (qx, qy)$). Hinweis: Sei $ax + by + c = 0$ die Koordinatenform der Geraden, die durch die Punkte P, Q definiert ist. Dann ist $d = |ax' + by' + c| / \sqrt{a^2 + b^2}$ der Abstand eines Punktes (x', y') von dieser Geraden. (8 Punkte)
- (c) Sei ein Rechteck durch seine linke untere Ecke (lx, ly) und seine rechte obere Ecke (rx, ry) definiert. Schreiben Sie eine Funktion `int rectangle(double x, double y, double lx, double ly, double rx, double ry)` die eine 1 liefert, falls der Abstand zwischen dem Punkt $X = (x, y)$ und einer der vier Seiten eines so definierten Rechtecks kleiner 0.5 und sonst 0. (5 Punkte)
- (d) Nutzen Sie die Funktionen um ein Programm zu schreiben, das folgende Ausgabe erzeugt: (15 Punkte)

```

24|                *****                X |
23|                *   *                  |
22|                *   *                  |
21|                *   *                  |
20|                *****                |
19|                *                       * |
18|                *                       * |
17|                *                       * |
16|                *                       * |
15|                *****                |
14|                *                       * |
13|                *                       * |
12|                *   *****                * |
11|                * *   *   *   *   *   * |
10|                * *   *   *   *   *   * |
9|                * *   *   *   *   *   * |
8|                * *   *   *   *   *   * |
7|                * *   *   *   *   *   * |
6|                * *   *   *   *   *   * |
5|                *   ***** *   ***** * |
4|                *                       *   * |
3|                *                       *   * |
2|                *                       *   * |
1|                *                       *   * |
y=0|X *****                |
-----
x=  012345678911111111112222222222333333
    01234567890123456789012345
Hinweise:
- X, |, und - gehören nicht zur Ausgabe und dienen nur zur Orientierung und Abgrenzung
- Das X unten-links hat die Koordinaten (0,0); das X oben-rechts (35,24)
- Die X-Koordinaten für > 9 sind zweireihig dargestellt

```

Hinweis: Die Schwierigkeit hier besteht darin, sich zu überlegen, wie das Ergebnis von Funktionen wie `line` oder `rectangle` genutzt werden kann, um Ausgabezeichen an der richtigen Stelle zu erzeugen. Nutzen Sie keine Arrays, keine globalen Variablen und modifizieren Sie nicht die Parameter der o.g. Funktion.