

Aufgabenblatt 2 - Programmierung in C

Vorpraktikum Informatik 2024

Letztes Update des Aufgabenblattes: 3. September 2024

Einleitung

Willkommen zum Aufgabenblatt für das Programmieren in C. In diesem Blatt findest du eine Reihe von Aufgaben, die dir helfen sollen, deine Programmierkenntnisse in C zu vertiefen.

Frage 1: If-Bedingungen

Aufgabe: Schreibe ein C-Programm, das eine Zahl einliest und überprüft, ob sie positiv, negativ oder null ist. Verwende dafür eine 'if'-Bedingung.

- a) Schreibe ein Programm, das eine Zahl vom Benutzer einliest und überprüft, ob diese Zahl positiv ist. Falls ja, gib „Die Zahl ist positiv“ aus.
- b) Modifiziere das Programm, sodass es zusätzlich überprüft, ob die Zahl negativ ist, und entsprechend „Die Zahl ist negativ“ ausgibt. Verwende dazu eine 'if-else' Struktur.
- c) Füge eine weitere Bedingung hinzu, die überprüft, ob die Zahl gleich null ist, und gib in diesem Fall „Die Zahl ist null“ aus. Verwende dazu eine 'if-else if-else' Struktur.

Hinweise:

- Achte darauf, die richtige Bedingung für jede Situation zu wählen.
- Denke daran, dass die Reihenfolge der Bedingungen wichtig ist, um unerwünschte Ergebnisse zu vermeiden.

Frage 2: Alterskategorie mit ‘else if‘

Aufgabe: Schreibe ein C-Programm, das das Alter des Benutzers einliest und basierend auf dem Alter eine entsprechende Kategorie ausgibt.

- a) Erstelle ein Programm, das das Alter einliest und überprüft, ob die Person ein Kind (unter 13 Jahre) ist. Gib „Kind“ aus, wenn die Bedingung erfüllt ist.
- b) Modifiziere das Programm, um zusätzlich zu prüfen, ob die Person ein Teenager (13 bis 19 Jahre) ist. Verwende dazu eine ‘else if‘ Bedingung und gib „Teenager“ aus, wenn die Bedingung erfüllt ist.
- c) Erweitere das Programm, um auch Erwachsene (20 Jahre und älter) zu berücksichtigen. Verwende eine ‘else‘ Bedingung, um „Erwachsener“ auszugeben, wenn keine der vorherigen Bedingungen zutrifft.

Hinweise:

- Verwende logische Operatoren wie ‘&&‘, um Bereiche abzudecken (z.B. für Teenager).
- Teste das Programm mit verschiedenen Altersangaben, um sicherzustellen, dass jede Kategorie korrekt erfasst wird.

Frage 3: Wochentage mit ‘switch‘

Aufgabe: Schreibe ein C-Programm, das eine Zahl zwischen 1 und 7 einliest und den entsprechenden Wochentag ausgibt.

- a) Erstelle ein Programm, das eine Zahl einliest und mit einer ‘switch‘-Anweisung überprüft, welcher Wochentag dieser Zahl entspricht (1 = Montag, 2 = Dienstag, usw.). Gib den entsprechenden Wochentag aus.
- b) Füge dem Programm einen ‘default‘-Fall hinzu, der eine Fehlermeldung ausgibt, falls die eingegebene Zahl nicht zwischen 1 und 7 liegt.
- c) Erweitere das Programm so, dass es auch einen zusätzlichen Hinweis für das Wochenende ausgibt, wenn der Wochentag Samstag (6) oder Sonntag (7) ist.

Hinweise:

- Achte darauf, dass jeder 'case'-Block mit einem 'break' endet, um ungewolltes Durchfallen (fall-through) zu vermeiden.
- Der 'default'-Fall ist nützlich, um unerwartete Eingaben zu behandeln.

Frage 4: Array-Deklaration und Zugriff

Aufgabe: Schreibe ein C-Programm, das ein Array von 10 ganzen Zahlen deklariert und initialisiert. Das Programm soll anschließend die Werte des Arrays in umgekehrter Reihenfolge ausgeben.

- a) Deklariere und initialisiere ein Array mit den Werten von 1 bis 10.
- b) Schreibe eine Schleife, die das Array durchläuft und die Werte in umgekehrter Reihenfolge auf dem Bildschirm ausgibt.

Hinweise:

- Verwende eine 'for'-Schleife, um das Array in umgekehrter Reihenfolge zu durchlaufen.

Frage 5: Berechnung des Durchschnitts mit Arrays

Aufgabe: Schreibe ein C-Programm, das den Benutzer auffordert, 5 Zahlen einzugeben, diese in einem Array speichert und den Durchschnitt der eingegebenen Zahlen berechnet und ausgibt.

- a) Deklariere ein Array, um 5 Ganzzahlen zu speichern.
- b) Lies die 5 Zahlen vom Benutzer ein und speichere sie im Array.
- c) Berechne den Durchschnitt der Zahlen und gib das Ergebnis aus.

Hinweise:

- Verwende eine 'for'-Schleife, um die Eingaben zu lesen und die Summe zu berechnen.
- Achte darauf, die Summe durch die Anzahl der Elemente zu teilen und eine Typumwandlung vorzunehmen.

Frage 6: Höchster und niedrigster Wert im Array

Aufgabe: Schreibe ein C-Programm, das ein Array von 10 Zahlen einliest und den höchsten und niedrigsten Wert im Array findet und ausgibt.

- a) Deklariere ein Array für 10 Ganzzahlen.
- b) Lies die 10 Zahlen vom Benutzer ein und speichere sie im Array.
- c) Durchlaufe das Array, um den höchsten und den niedrigsten Wert zu finden.
- d) Gib den höchsten und den niedrigsten Wert aus.

Hinweise:

- Verwende die ersten Elementwerte zur Initialisierung von 'max' und 'min'.
- Durchlaufe das Array, um die höchsten und niedrigsten Werte zu bestimmen.

Frage 7: Summieren von Array-Elementen

Aufgabe: Schreibe ein C-Programm, das ein Array von 8 Zahlen einliest und die Summe aller Elemente im Array berechnet und ausgibt.

- a) Deklariere ein Array für 8 Ganzzahlen.
- b) Lies die 8 Zahlen vom Benutzer ein.
- c) Berechne die Summe der Zahlen im Array.
- d) Gib die berechnete Summe aus.

Frage 8: Multiplikationstabelle mit Arrays

Aufgabe: Schreibe ein C-Programm, das eine Multiplikationstabelle für die Zahlen 1 bis 10 erstellt und in einem zweidimensionalen Array speichert. Gib die Tabelle anschließend formatiert aus.

- a) Deklariere ein zweidimensionales Array für die Multiplikationstabelle.
- b) Fülle das Array mit den Produkten der entsprechenden Zahlen.
- c) Gib die Tabelle formatiert aus.

Hinweise:

- Verwende verschachtelte Schleifen, um das Array zu füllen und auszugeben.
- Achte auf eine saubere Formatierung der Ausgabe.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
4	8	12	16	20	24	28	32	36	40
4	8	12	16	20	24	28	32	36	40
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Abbildung 1: Beispielhafte Multiplikationstabelle von 1 bis 10, gespeichert in einem zweidimensionalen Array.

Frage 9: Verwendung von Funktionen

Aufgabe: Schreibe ein C-Programm, das eine einfache mathematische Funktion implementiert und aufruft.

- a) Erstelle eine Funktion, die das Quadrat einer Zahl berechnet und das Ergebnis zurückgibt.
- b) Schreibe ein Programm, das den Benutzer auffordert, eine Zahl einzugeben, und das Ergebnis der Quadratberechnung ausgibt.
- c) Füge eine zweite Funktion hinzu, die die Kubikzahl der Zahl berechnet und in ähnlicher Weise ausgibt.

Frage 10: Lokale, globale und statische Variablen

Aufgabe: Untersuche das Verhalten von lokalen, globalen und statischen Variablen in einem C-Programm.

- a) Erstelle ein Programm mit einer globalen Variablen, die von mehreren Funktionen gelesen und geändert wird.
- b) Implementiere eine Funktion, die eine lokale Variable verwendet und ihre Gültigkeit innerhalb und außerhalb der Funktion testet.
- c) Verwende eine statische Variable innerhalb einer Funktion und zeige, wie ihr Wert über mehrere Funktionsaufrufe hinweg beibehalten wird.

Frage 11: Verwenden von Arrays mit Funktionen

Aufgabe: Schreibe ein C-Programm, das ein Array an eine Funktion übergibt und die Elemente des Arrays verarbeitet.

- a) Erstelle eine Funktion, die die Summe der Elemente eines Arrays berechnet und zurückgibt.
- b) Schreibe ein Programm, das ein Array von Ganzzahlen deklariert, füllt und die Summe der Elemente berechnet.
- c) Implementiere eine zweite Funktion, die das Maximum der Array-Elemente findet und zurückgibt.

Frage 12: Daten aus Funktionen zurückgeben

Aufgabe: Schreibe ein C-Programm, das eine Funktion verwendet, um das Ergebnis einer Berechnung zurückzugeben.

- a) Implementiere eine Funktion, die den größten gemeinsamen Teiler (GGT) von zwei Zahlen berechnet und zurückgibt.
- b) Schreibe ein Programm, das zwei Zahlen vom Benutzer einliest und den GGT der beiden Zahlen berechnet und ausgibt.

- c) Erweitere das Programm, um den Benutzer wiederholt nach Zahlen zu fragen und den GGT zu berechnen, bis der Benutzer das Programm beendet.

Frage 13: Aufrufen von Funktionen innerhalb anderer Funktionen & einfache Rekursion

Aufgabe: Erkunde das Konzept der Rekursion in C und das Aufrufen von Funktionen innerhalb anderer Funktionen. Schreibe ein Programm, das diese Funktionen verwendet und die Ergebnisse ausgibt.

- a) Erstelle eine Funktion, die die Fakultät einer Zahl berechnet, indem sie sich selbst rekursiv aufruft.
- b) Implementiere eine Funktion, die eine andere Funktion aufruft, um die Summe der ersten n natürlichen Zahlen zu berechnen.

Frage 14: Dateien lesen und schreiben

Aufgabe: Schreibe ein C-Programm, das eine Textdatei erstellt, Daten in diese schreibt und die Daten anschliessend wieder ausliest und ausgibt.

- a) Erstelle ein Programm, das eine Textdatei namens 'output.txt' öffnet und eine Reihe von Zeichenketten in diese Datei schreibt.
- b) Implementiere eine Funktion, die die Datei 'output.txt' öffnet und den gesamten Inhalt der Datei auf dem Bildschirm ausgibt.
- c) Füge dem Programm eine Option hinzu, die es dem Benutzer ermöglicht, entweder neue Daten in die Datei zu schreiben oder den vorhandenen Inhalt zu lesen.

Frage 15: Anhaengen an eine Datei

Aufgabe: Schreibe ein C-Programm, das es ermöglicht, Daten an das Ende einer bestehenden Datei anzuhängen.

- a) Erstelle ein Programm, das eine Textdatei oeffnet und eine neue Zeile Text an das Ende der Datei anhaengt.
- b) Implementiere eine Funktion, die die gesamte Datei ausliest und den Inhalt auf dem Bildschirm anzeigt, nachdem neue Daten anhaengt wurden.
- c) Fuege eine Benutzeroption hinzu, um festzulegen, ob der Inhalt der Datei geloescht und neu geschrieben oder nur Daten anhaengt werden sollen.

Frage 16: Zeichenweise Dateioperationen und Zeichenfrequenzanalyse

Aufgabe: Schreibe ein C-Programm, das eine Datei zeichenweise liest, die Zeichen auf dem Bildschirm ausgibt und eine einfache Zeichenfrequenzanalyse durchfuehrt.

- a) Erstelle eine Funktion, die eine Datei oeffnet und zeichenweise einliest. Implementiere die Funktion so, dass jedes Zeichen, das aus der Datei gelesen wird, auf dem Bildschirm ausgegeben wird.
- b) Fuege eine Bedingung hinzu, die nur alphabetische Zeichen ausgibt und alle anderen Zeichen ignoriert.
- c) Implementiere eine zusaezliche Funktion, die die Haeufigkeit jedes Buchstabens im Text zaehlt und diese Haeufigkeiten am Ende der Dateioperation auf dem Bildschirm ausgibt.

Hinweise:

- Nutze ein Array oder eine andere Datenstruktur, um die Haeufigkeit der Buchstaben (a-z, A-Z) zu speichern.
- Achte darauf, Groeß- und Kleinbuchstaben zusammen zu zaehlen (z.B. A und a sollten als gleiches Zeichen betrachtet werden).

Bonusaufgaben

Bonusfrage 1: Zeilenweise Dateioperationen und Textmanipulation

Aufgabe: Schreibe ein C-Programm, das eine Datei zeilenweise liest, den Inhalt der Datei auf dem Bildschirm ausgibt und bestimmte Textmanipulationen durchführt.

- a) Erstelle eine Funktion, die eine Datei öffnet und zeilenweise einliest. Implementiere die Funktion so, dass jede Zeile, die aus der Datei gelesen wird, auf dem Bildschirm ausgegeben wird.
- b) Ergänze das Programm um eine Möglichkeit, nur die ersten n Zeilen der Datei zu lesen und auszugeben, wobei n vom Benutzer festgelegt wird.
- c) Implementiere eine Funktion, die alle Vorkommen eines bestimmten Worts in der Datei durch ein anderes Wort ersetzt, und speichere das Ergebnis in einer neuen Datei.

Hinweise:

- Implementiere eine Benutzerabfrage, um das zu ersetzende Wort und das Ersatzwort einzugeben.
- Achte darauf, dass die neue Datei nicht die Originaldatei überschreibt, sondern als separate Datei gespeichert wird.
- Überprüfe, wie du die Suche nach dem Wort und die Ersetzung effizient gestalten kannst (z.B. durch den Einsatz von `strstr` und `strcpy`).

Bonusfrage 2: Array kopieren und umkehren

Aufgabe: Schreibe ein C-Programm, das ein vorgegebenes Array kopiert, beide Arrays ausgibt und das kopierte Array anschließend rückwärts ausgibt.

- a) Erstelle ein Programm, das ein Array mit festen Werten deklariert und ein zweites Array der gleichen Größe erstellt.
- b) Kopiere die Werte des ersten Arrays in das zweite Array und gib beide Arrays in der ursprünglichen Reihenfolge aus.

- c) Erweitere das Programm so, dass das zweite Array rückwärts ausgegeben wird.

Bonusfrage 3: Funktion zur Berechnung des Mindestwertes in einem Array

Aufgabe: Schreibe ein C-Programm, das den kleinsten Wert in einem Array von Ganzzahlen findet.

- a) Erstelle eine Funktion, die den kleinsten Wert in einem Array berechnet und zurückgibt.
- b) Implementiere ein Programm, das ein Array deklariert, füllt und die Funktion aufruft, um den kleinsten Wert zu ermitteln.
- c) Erweitere das Programm so, dass der Benutzer die Größe des Arrays und die Elemente eingeben kann.

Bonusfrage 4: Datei mit Benutzereingaben erstellen

Aufgabe: Schreibe ein C-Programm, das eine Datei erstellt und Daten basierend auf Benutzereingaben speichert.

- a) Erstelle ein Programm, das den Benutzer auffordert, Text einzugeben und diesen in einer Datei speichert.
- b) Implementiere eine Funktion, die die Datei ausliest und den Inhalt auf dem Bildschirm anzeigt.
- c) Füge eine Option hinzu, um den Benutzereingaben weitere Zeilen hinzuzufügen oder die Datei zu überschreiben.

Bonusfrage 5: Suchen eines Wertes in einem Array

Aufgabe: Schreibe ein C-Programm, das nach einem bestimmten Wert in einem Array sucht.

- a) Erstelle eine Funktion, die einen Wert in einem Array sucht und den Index des Wertes zurückgibt.

- b) Implementiere ein Programm, das ein Array deklariert, füllt und die Suchfunktion aufruft.
- c) Erweitere das Programm, um den Benutzer nach dem zu suchenden Wert zu fragen.

Bonusfrage 6: Zeichenumkehr in einer Datei

Aufgabe: Schreibe ein C-Programm, das den Inhalt einer Textdatei zeichenweise liest und den umgekehrten Inhalt in eine neue Datei schreibt.

- a) Erstelle eine Funktion, die den Inhalt einer Datei liest und in einem Array speichert.
- b) Implementiere eine Funktion, die das Array umkehrt und den Inhalt in eine neue Datei schreibt.
- c) Füge eine Option hinzu, um die umgekehrte Datei auf dem Bildschirm anzuzeigen.

Bonusfrage 7: Durchschnittsberechnung mit einer dynamischen Arraygröße

Aufgabe: Schreibe ein C-Programm, das den Durchschnitt von Zahlen berechnet, wobei die Größe des Arrays zur Laufzeit festgelegt wird.

- a) Erstelle ein Programm, das den Benutzer nach der Anzahl der Zahlen fragt und dann Speicher für das Array dynamisch allokiert.
- b) Implementiere eine Funktion zur Berechnung des Durchschnitts der Array-Elemente.
- c) Füge eine Option hinzu, um das Array freizugeben, nachdem die Berechnungen abgeschlossen sind.

Bonusfrage 8: Entfernung von Leerzeichen in einer Datei

Aufgabe: Schreibe ein C-Programm, das Leerzeichen aus einer Datei entfernt und den bearbeiteten Text in eine neue Datei schreibt.

- a) Erstelle eine Funktion, die eine Datei zeichenweise liest und Leerzeichen ignoriert.
- b) Implementiere die Funktion so, dass der bearbeitete Text in eine neue Datei geschrieben wird.
- c) Füge eine Option hinzu, die es dem Benutzer ermöglicht, die Original- und die bearbeitete Datei zu vergleichen.

Bonusfrage 9: Zeilenweise Verarbeitung einer Datei

Aufgabe: Schreibe ein C-Programm, das die Anzahl der Zeichen, Wörter und Zeilen in einer Textdatei zählt.

- a) Erstelle eine Funktion, die die Anzahl der Zeichen in einer Datei zählt.
- b) Implementiere Funktionen, die die Anzahl der Wörter und Zeilen in der Datei zählen.
- c) Füge eine Option hinzu, um die Ergebnisse der Zählungen auf dem Bildschirm auszugeben.

Bonusfrage 10: Sortieren eines Arrays mit einer Funktion

Aufgabe: Schreibe ein C-Programm, das ein Array sortiert. Verwende dazu eine eigene Sortierfunktion.

- a) Erstelle eine Funktion, die ein Array von Ganzzahlen sortiert.
- b) Implementiere ein Programm, das ein Array deklariert, füllt und die Sortierfunktion aufruft.
- c) Füge eine Option hinzu, um das Array vor und nach dem Sortieren auszugeben.

Bonusfrage 11: Kopieren von Dateien

Aufgabe: Schreibe ein C-Programm, das den Inhalt einer Datei in eine andere Datei kopiert.

- a) Erstelle eine Funktion, die eine Datei zeichenweise liest und in eine neue Datei schreibt.
- b) Implementiere das Programm so, dass der Benutzer die Quelldatei und die Zieldatei angeben kann.
- c) Füge eine Überprüfung hinzu, um sicherzustellen, dass die Quelldatei erfolgreich geöffnet wurde.

Bonusfrage 12: Ermitteln der Dateigröße

Aufgabe: Schreibe ein C-Programm, das die Größe einer Datei in Bytes ermittelt und auf dem Bildschirm ausgibt.

- a) Erstelle eine Funktion, die die Größe einer Datei berechnet.
- b) Implementiere das Programm so, dass der Benutzer den Dateinamen eingeben kann.
- c) Füge eine Überprüfung hinzu, um sicherzustellen, dass die Datei erfolgreich geöffnet wurde.

Bonusfrage 13: Verwaltung von Studierendendaten mit Arrays

Aufgabe: Schreibe ein C-Programm, das die Namen und Noten von Studierenden speichert und die Durchschnittsnote berechnet.

- a) Erstelle ein Programm, das die Namen und Noten einer Gruppe von Studierenden speichert.
- b) Implementiere eine Funktion, die die Durchschnittsnote berechnet und ausgibt.
- c) Füge eine Option hinzu, um die Daten eines bestimmten Studierenden abzufragen.

Bonusfrage 14: Erstellung und Verwaltung eines einfachen Menüs

Aufgabe: Schreibe ein C-Programm, das ein einfaches Textmenü erstellt, das dem Benutzer mehrere Optionen bietet.

- a) Erstelle ein Menü, das Optionen wie „Neue Datei erstellen“, „Datei lesen“ und „Programm beenden“ anbietet.
- b) Implementiere Funktionen, die die entsprechenden Aktionen ausführen.
- c) Füge eine Schleife hinzu, die das Menü anzeigt, bis der Benutzer das Programm beendet.

Bonusfrage 15: Suche und Ersetzung in einer Datei

Aufgabe: Schreibe ein C-Programm, das nach einem bestimmten Wort in einer Datei sucht und es durch ein anderes Wort ersetzt.

- a) Erstelle eine Funktion, die eine Datei nach einem bestimmten Wort durchsucht.
- b) Implementiere eine Funktion, die das gefundene Wort durch ein neues ersetzt.
- c) Füge eine Benutzeroption hinzu, um das ursprüngliche und das neue Wort einzugeben.

Bonusfrage 16: Zählen von spezifischen Zeichen in einer Datei

Aufgabe: Schreibe ein C-Programm, das die Anzahl eines bestimmten Zeichens in einer Datei zählt.

- a) Erstelle eine Funktion, die eine Datei öffnet und die Anzahl eines bestimmten Zeichens zählt.
- b) Implementiere das Programm so, dass der Benutzer das Zeichen eingeben kann.
- c) Gib die Anzahl des Zeichens auf dem Bildschirm aus.

Bonusfrage 17: Arbeiten mit zweidimensionalen Arrays

Aufgabe: Schreibe ein C-Programm, das mit einem zweidimensionalen Array arbeitet und die Summe jeder Zeile und Spalte berechnet.

- a) Erstelle ein zweidimensionales Array und fülle es mit Zufallszahlen.
- b) Implementiere Funktionen, die die Summe jeder Zeile und jeder Spalte berechnen.
- c) Gib die Summen der Zeilen und Spalten auf dem Bildschirm aus.

Bonusfrage 18: Zufallszahlen in einer Datei speichern

Aufgabe: Schreibe ein C-Programm, das eine bestimmte Anzahl von Zufallszahlen generiert und diese in einer Datei speichert.

- a) Erstelle eine Funktion, die eine Datei öffnet und eine bestimmte Anzahl von Zufallszahlen darin speichert.
- b) Implementiere das Programm so, dass der Benutzer die Anzahl der Zufallszahlen eingeben kann.
- c) Füge eine Funktion hinzu, die die Zufallszahlen aus der Datei liest und auf dem Bildschirm ausgibt.