

Programmieren Tutorium - Arbeitsblatt 4

Alle ungeraden Monats Nummern ausgeben

Schreibe ein Programm, das alle ungeraden Monats Nummern (1 bis 12) in absteigender Reihenfolge ausgibt. Das Ergebnis sollte so aussehen: 11, 9, 7, 5, 3, 1

Hinweis: Nutze eine Schleife

Übersetzen Sie in C#

a) Die Variable txt ist ein String. Für alle Zeichen des Strings sollen Minuszeichen (-)

durch Pluszeichen (+) ersetzt werden.

b) Sie haben ein Feld(Array) vector von ganzen Zahlen und eine weitere ganze Zahl

summe. Jedes dritte Element des Feldes soll zu der Zahl summe addiert werden

Erstelle eine Funktion “isEven” / “istGerade”

- a) Erstelle eine Funktion die prüft ob eine ganze Zahl gerade ist. Die Funktion muss ein bool zurückgeben.
- b) Erstelle wieder eine Funktion die prüft ob eine ganze Zahl gerade ist. Jetzt soll die Funktion, aber nichts zurückgeben sonder Ausgeben: “x ist gerade” oder “x ist ungerade”

Hinweis: Nutze modulo %

Erstelle eine Funktion, die prüft ob eine Zahl durch 3 teilbar, aber ungerade ist

Erstelle eine Funktion, die prüft, ob eine Zahl durch 3 teilbar ist, also kein Rest bei Dividierung durch z.B 3, 6, 9, 12. Gebe ein bool zurück;

Also istUngeradeUndDurchDreiTeilbar(int x)

istUngeradeUndDurchDreiTeilBar(2) => false

istUngeradeUndDurchDreiTeilBar(3) => true

istUngeradeUndDurchDreiTeilBar(6) => false

istUngeradeUndDurchDreiTeilBar(9) => true

Hinweis: Nutze modulo %

Ausgabe eines Musters mit verschachtelten Schleifen

Aufgabe: Erstelle ein Programm, das ein Muster von Hashtags auf der Konsole ausgibt. Das Muster soll ein Dreieck darstellen, bei dem jede Zeile einen Hashtag mehr enthält als die vorherige Zeile.

Beispiel:

#

##

###

Hinweis:

Nutze eine verschachtelte **for**-Schleife, um in jeder Zeile die entsprechende Anzahl an Sternen auszugeben.

Zweitkleinstes Element in einem Array finden

Schreibe eine Funktion **ZweitkleinstesElement(int[] array)**, die das zweitkleinste Element eines Arrays von Ganzzahlen zurückgibt. Gib eine Nachricht aus, falls das Array weniger als zwei Elemente enthält.

Dieses Werk von Tobias Bück ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz (CC BY 4.0)

Beispiel:

Eingabe: [4, 2, 7, 1, 9]

Ausgabe: 2

Hinweis: Verwende Variablen, um das kleinste und zweitkleinste Element zu speichern und vergleiche bei jedem Element im Array.

Durchschnitt der Zahlen über einem Schwellenwert berechnen

Schreibe eine Funktion `DurchschnittSchwelle(int[] array, int schwelle)`, die den Durchschnitt der Zahlen berechnet, die über einem bestimmten Schwellenwert liegen.

Beispiel:

Eingabe: Array = [10, 20, 30, 40], Schwelle = 15

Ausgabe: 30 // da nur 20, 30 und 40 größer als 15 sind und deren Durchschnitt 30 ist.

Hinweis: Summiere nur die Zahlen, die den Schwellenwert überschreiten, und zähle sie, um am Ende den Durchschnitt zu berechnen.

Produkt einer Zahlenreihe berechnen

Aufgabe: Schreibe eine Funktion `BerechneProdukt(int n)`, die das Produkt der Zahlen von 1 bis `n` berechnet und das Ergebnis zurückgibt. Wenn `n` negativ oder 0 ist, soll die Funktion 1 zurückgeben.

Beispiel:

Eingabe: 4

Ausgabe: 24 // da $1 * 2 * 3 * 4 = 24$

Hinweis: Beginne die Multiplikation bei 1, und überprüfe die Eingabe, um sicherzustellen, dass das Produkt für `n <= 0` immer 1 ist.

Anzahl der positiven und negativen Zahlen zählen

Aufgabe: Schreibe eine Funktion `ZaehleVorzeichen(int[] array)`, die durch ein Array von Ganzzahlen geht und die Anzahl der positiven und negativen Zahlen zählt. Das Ergebnis soll als Nachricht ausgegeben werden.

Beispiel:

Eingabe: [-1, 4, -5, 2, 0, 3]

Ausgabe: 3 positive Zahlen, 2 negative Zahlen

Hinweis: Prüfe mit `zahl > 0` auf positive Zahlen und mit `zahl < 0` auf negative Zahlen.

Suche nach einem Element und gebe die Fundstellen aus

Aufgabe: Erstelle ein Programm, das nach allen Positionen eines bestimmten Wertes in einem Array von Wörtern sucht und die Positionen ausgibt. Wenn der Wert nicht gefunden wird, soll eine Nachricht erscheinen, dass der Wert nicht enthalten ist.

Beispiel:

Eingabe: Array = {"Apfel", "Banane", "Apfel", "Kirsche"}, Suche = "Apfel"

Ausgabe: Gefunden bei Position: 0, 2

Hinweis: Verwende eine Schleife und speichere alle gefundenen Positionen in einer Liste, falls der Wert mehrfach vorkommt.

Sortieren eines Arrays mit Auswahl-Algorithmus (Selection Sort)

Schreibe eine Funktion `SelectionSort(int[] array)`, die ein Array von Zahlen mit dem Auswahl-Algorithmus (Selection Sort) sortiert und zurückgibt.

Information zu Selection Sort:

Der Algorithmus sucht in jedem Durchlauf das kleinste Element und tauscht es mit dem Element an der aktuellen Position. Das Array wird so von vorne nach hinten schrittweise sortiert.

Beispiel für die Schritte:

Anfang: [4, 2, 7, 1, 3]

Nach erstem Durchlauf: [1, 2, 7, 4, 3]

usw.

Beispiel für (a):

Eingabe: [3, 1, 4, 2]

Ausgabe: [1, 2, 3, 4]

Aufgabe (b): Erstelle eine Funktion `Umkehren(int[] array)`, die das sortierte Array in umgekehrter Reihenfolge ausgibt.

Beispiel für (b):

Eingabe: [1, 2, 3, 4]

Ausgabe: [4, 3, 2, 1]

Hinweis: Verwende zwei separate Funktionen für Sortierung und Umkehrung, und vermeide eingebaute Sortier- oder Umkehrfunktionen, um den Algorithmus manuell zu üben.