

Kreditkarten werden durch sechszehnziffrige Ziffernfolgen identifiziert. Die ersten fünfzehn Ziffern werden durch die Bank vergeben, die letzte dient als Prüfziffer, die aus den anderen fünfzehn Ziffern berechnet wird. Sie wird in drei Schritten ermittelt. Wir nehmen an, dass die Kreditkarte die Ziffernfolge 271 828 182 845 856 ohne Prüfziffer enthält. Wir schreiben diese Ziffern in eine Tabelle:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	7	1	8	2	8	1	8	2	8	4	5	8	5	6

1. Wir verdoppeln die Ziffern an den Stellen mit ungerader Nummer und addieren sie **ziffernweise**. Im Beispiel:

$$4 + 2 + 4 + 2 + 4 + 8 + (1 + 6) + (1 + 2) = 34$$

2. Die Ziffern an den Stellen mit gerader Nummer werden addiert. Im Beispiel:

$$7 + 8 + 8 + 8 + 8 + 5 + 5 = 49$$

3. Die beiden Summen werden addiert und dann die Differenz zur nächsten Zehnerzahl ermittelt. Diese Differenz ist dann die Prüfziffer. Im Beispiel:

$$34 + 49 = 83; \quad 90 - 83 = 7$$

Die um die 7 ergänzte Ziffernfolge ergibt nun die Kreditkartennummer 2718 2818 2845 8567.

Ähnliche Prüfzifferberechnungen gibt es für den EAN-Code auf der Milchtüte, die ISBN-Nummer auf einem Buch, die Seriennummer auf dem Euro-Schein oder die Pharmazentralnummer auf der Medikamentenverpackung. Mit diesen Prüfziffern sollen Fehler bei der Eingabe erkannt werden. Sie werden überall dort benötigt, wo sich an die Datenerfassung kein zusätzlicher Prüfgang anschließt.

Aufgabe: Schreibe ein Programm, das zu gegebener Ziffernfolge die Prüfziffer ermittelt.

Tipps:

- Mit `int x`; sei eine Integer-Variable deklariert. Durch die Zuweisung `x = 13/4`; erhält die Variable `x` den Wert 3, d.h. das ganzzahlige Ergebnis der Division ohne Rest.
- Mit `int x`; sei eine Integer-Variable deklariert. Durch die Zuweisung `x = 13%4`; erhält die Variable `x` den Wert 1, d.h. den Rest bei der ganzzahligen Division.
- Die Ziffernfolge soll in einem Array gespeichert werden.