



Liebe Schülerin, lieber Schüler,

wir freuen uns, dich heute zum Sächsischen Informatikwettbewerb begrüßen zu können und wünschen dir viel Erfolg, aber auch Freude bei der Lösung der Aufgaben.

Hier noch einige Hinweise:

Arbeitszeit

- Für die Lösung der Aufgaben hast du 2,5 Stunden (150 min) Zeit.

Hilfsmittel

- Als Hilfsmittel sind Standardsoftware (Textverarbeitung, Tabellenkalkulation, Datenbanken) sowie Taschenrechner und Tafelwerk zugelassen.
- Über die zugelassenen Programmiersysteme informiert dich dein Lehrer.

Bewertung

- Für die Aufgabe 1 gibt es 10 Punkte, für die Aufgabe 2 werden 20 Punkte vergeben.
- **Zu jeder Aufgabe ist ein Teil der Aufgaben auf dem Papier zu lösen. Beachte dazu auch die Punktverteilung auf den Aufgabenzetteln.**
- Es ist wichtig, dass der Lösungsweg deutlich wird.

Wir wünschen dir viel Erfolg!

Sicherheit geht vor

Endlich gibt es vor dem Tresorraum der Bank zusätzlich zum Fingerabdruck-Scanner noch eine Zugangssicherung, die mit einem vierstelligen Zahlencode funktioniert.

Der Bankdirektor gibt zum Start die Zahl 1986 vor. Aus Sicherheitsgründen wird diese Zahl jeden Tag wie folgt geändert:

- die Einerstelle der Quersumme der 4 Ziffern wird als neue Einerstelle rechts angefügt,
- dafür wird die linksstehende Ziffer gestrichen.

Aus 1986 wird demnach 9864, weil $\text{Quersumme}(1986) = 1 + 9 + 8 + 6 = 24$

Der Bankdirektor muss noch 1611 Tage arbeiten. Er möchte wissen, wann bei diesem Verfahren in seiner verbleibenden Arbeitszeit der Ausgangscode wieder erreicht wird.

Aufgaben:

- Gib die Codes für die ersten 5 Tage. 2 Punkte
- Erstelle eine Computerlösung, die aus einem Code den nächsten Code erzeugt. 5 Punkte
- Erstelle eine Computerlösung, die solange Codes erzeugt, bis der Startcode wieder erreicht ist. Dann soll ausgegeben werden, wie viele Tage nötig waren. 3 Punkte

Paritätsbit

Nichts wird dem Zufall überlassen. Stell dir vor, deine Eltern wollen dir zur Jugendweihe eine große Summe Geld überweisen, doch bei der Übertragung der Kontodaten geht etwas schief - wie ärgerlich. Zum Glück überlässt man dabei nichts dem Zufall.

Wie du sicher weißt, werden Daten als Binärcode übertragen. Am Ende dieses Binärcodes gibt es ein sogenanntes Prüfbit - auch Paritätsbit genannt. Ziel dieses Bits ist es, die Anzahl der 1-en gerade zu halten.

Aufgaben:

- a) Gib für das folgende Bitmuster das anzuhängende Paritätsbit an: 011110001111 1 Punkt
- b) Gegeben ist das Bitmuster 0011010x11 und das Prüfbit 1. An der Stelle x wurde leider kein entsprechendes Bit empfangen. Gib das fehlende Bit an und begründe. 2 Punkte
- c) Unter welcher Voraussetzung funktioniert die Erkennung verfälschter Bits nicht? Begründe deine Aussage. 1 Punkt
- d) Erstelle eine Computerlösung, die zu einem gegebenen Bitmuster der Länge 8 das zugehörige Paritätsbit ermittelt. 2 Punkte

Ein weiteres Verfahren kontrolliert die Übertragung des Binärcodes durch zeilenweises Einfügen in eine entsprechende Matrix (Tabellenform). Die Kontrolle erfolgt durch Paritätsbits in Zeilen und Spalten. In der Abbildung wird das für die Nachricht 0010110101111111 dargestellt.

0	0	1	0	1
1	1	0	1	1
0	1	1	1	1
1	1	1	1	0
0	1	1	1	

- e) Du empfangst die Nachricht 0100001100000101, bei der ein Bit falsch übertragen wurde. Als zeilenweise gebildete Prüfbits wurde 1100 übertragen, als spaltenweise gebildete Prüfbits 0000. Beschreibe das Verfahren zum Finden des fehlerhaften Bits. Gib an, welches Bit fehlerhaft ist und nenne die korrekte Nachricht. 4 Punkte
- f) Erstelle eine Computerlösung, die mit Hilfe des oben beschriebenen Verfahrens ein gegebenes Bitmuster der Länge 16 mit Hilfe der 8 Paritätsbits auf Fehler überprüft. 4 Punkte
- g) Erweitere deine Computerlösung so, dass ein fehlerhaft übertragenes Bit automatisch korrigiert wird. 2 Punkte
- h) Vergleiche die beiden beschriebenen Verfahren bezüglich ihrer Fehlererkennung und Korrekturmöglichkeit. 2 Punkte

- i) Gegeben ist folgender Binärcode mit zugehörigen Paritätsbits. Genau ein Bit ist falsch übertragen worden. Welches ist das falsche Bit? Begründe deine Aussage. 2 Punkte

0	0	1	0	1
0	0	1	0	1
0	0	0	0	0
0	0	1	0	1
0	1	1	0	