

Übung 8

Algorithmen und Programmierung

Jonas Henschel, Jens Pönisch, Dominik Gorgosch, Bastian Felix Bachmann, Arvid Horn, Billy Naumann

Aufgabe 1 (Bitoperatoren)

- a) Schreiben Sie eine Funktion, die den Wert des n -ten Bits (0 oder 1) eines übergebenen Integers zurückliefert. Das 0-te Bit ist das Least Significant Bit.
- b) Schreiben Sie eine Funktion, die das n -te Bit eines Integers auf 1 setzt.
- c) Schreiben Sie eine Funktion, die das n -te Bit eines Integers auf 0 setzt.

Aufgabe 2 (Laufflichter)

Bald ist wieder Weihnachten und Sie möchten sich selbst und Ihrer Familie eine kleine Freude mit einer blinkenden Lichterkette machen.

- a) Um vorerst etwas herumexperimentieren zu können, simulieren sie die LEDs der Lichterkette als Bitstring. Sie schreiben dazu eine Funktion, welche eine übergebene Variable vom Typ **unsigned int** bitweise auf der Kommandozeile ausgibt.

```
1 Wert der Variable: 42
2 Ausgabe: 0000000000000000000000000101010
```

- b) Um sich warm zu machen, wollen Sie zunächst die gesamte Lichterkette blinken lassen. Dazu schalten Sie zuerst alle LEDs ein und dann wieder aus. Das wiederholen sie endlos.
- c) Als nächsten Schritt sollen die LEDs abwechselnd aufleuchten, sodass sich folgendes Muster ergibt:

```
1 Schritt 1: 01010101
2 Schritt 2: 10101010
3 Schritt 3: 01010101
4 Schritt 4: 10101010
5 usw...
```

- d) Als Sie das Blinken dieses Musters betrachten, erinnern Sie sich an die Laufflichter, welche häufig an Schaufenstern zu sehen sind. Sie wollen mit einem einfachen Muster beginnen. Deshalb schalten sie zuerst alle LEDs von einer Seite beginnend ein. Anschließend schalten sie die LEDs in der gleichen Reihenfolge wieder aus, in der sie eingeschaltet wurden.

```

1 Schritt 1: 00000001
2 Schritt 2: 00000011
3 Schritt 3: 00000111
4 usw...
5 Schritt 8: 11111111
6 Schritt 9: 11111110
7 Schritt 10: 11111100
8 Schritt 11: 11111000
9 usw...
10 Schritt 16: 00000000

```

- e) Ihr Lauflicht gefällt Ihnen bereits ganz gut. Allerdings wollen Sie auch verschiedene Muster durch die Kette laufen lassen. Der Einfachheit halber sollen die Muster durch eine Ganzzahl beschrieben werden können, welche durch Ihr Programm eingelesen wird. Dieses Muster soll dann “im Kreis” durch die Lichterkette laufen.

```

1 Muster: 42
2 Schritt 1: 00101010
3 Schritt 2: 01010100
4 Schritt 3: 10101000
5 Schritt 4: 01010001
6 Schritt 5: 10100010
7 usw...

```

Hinweise:

- Die Bibliothek `unistd.h` bietet ihnen die Funktion `usleep()` an. Mit dieser Funktion können Sie Ihr Programm für eine gewisse Anzahl Mikrosekunden unterbrechen.
- Mit dem Steuerzeichen `\r` können Sie mit dem Ausgabecursor wieder an den Anfang der Zeile springen.
- Auf manchen Systemen wird der Ausgabepuffer ohne abschließendes `\n` nicht auf das Terminal geschrieben. Mit der Funktion `fflush(stdout)` können sie das Schreiben auf das Terminal erzwingen.
- Die Beispiele sind ab Aufgabe c) auf 8 Bit gekürzt. Sie sollen weiterhin alle bits einer Variable vom Typ **unsigned int** verwenden.

Aufgabe 3 (Fremdcode)

Sie finden auf OPAL bei den Übungsblättern ein ZIP-Archiv mit drei C-Dateien.

1. Bitte lesen Sie sich den Code durch und versuchen Sie zu verstehen, was die Programme machen.
2. Ordnen Sie die Dateien nach ihrer Verständlichkeit.
3. Finden Sie Kriterien, welche den Code für sie besonders gut lesbar gemacht haben.