

Name(n):

eigentlich kontrollierende(r) Tutor(in) (Postfach):

Aufgabe 1:

Gegeben ist folgender Code:

```
function ausgabe = fkt(n)
ausgabe = '';
if n == 0
    ausgabe='0';
end
while n>0
    st=mod(n,2);
    ausgabe=[num2str(st), ausgabe];
    n=floor(n/2);
end
```

1. Beschreiben Sie in einem Satz, was das Programm macht!
2. Welche Ausgabe erhalten Sie bei dem Aufruf $fkt(27)$? (Tipp: Geben Sie ggf. **nur zur Kontrolle** das Programm in Matlab ein!)

Lösung:

1. Das Programm gibt die Zahl n in Binärdarstellung an.
2. $fkt(27) = 11011_2$.

Aufgabe 2:

Gegeben ist folgender Code:

```
1
2 function uebung(a,b)
3     x = [ ]
4     n = 0;
5     ende = a*b;
6
7     if a<b
8         c = a;
9         a = b;
10        b = c;
11    end
12
13    for k=a:ende
14        if (mod(k,a)==0 && mod(k,b)==0)
15            n = n + 1;
16            x(n) = k;
17        end
18    end
19
20    disp(x)
21 (Ende des Programms)
```

1. Wozu dient die if-Abfrage? Also was passiert dort und warum ist sie notwendig?
2. Welche Ausgabe liefert *uebung(10,15)*?

Lösung:

1. Die Eingabewerte werden umgedreht. Eigentlich ist sie nicht notwendig.
2. $uebung(10,15) = [30\ 60\ 90\ 120\ 150]$. Das Programm ermittelt also die gemeinsamen Vielfachen bis zum Ende ($a \cdot b = 10 \cdot 15 = 150$).

Aufgabe 3:

Gegeben ist ein fehlerhafter Code! Ohne Fehler rechnet das Programm die einzugebene Zahl n in eine beliebige Basis q um. Als Output erhält man einen Vektor z mit den Ziffern $(1, \dots, q-1)$ als Einträge.

```
1 function z = dec2basis(n)
2
3 if (n=0)
4     erg = 0;
5 else
6     erg = [ ];
7 end;
8
9 while (n=0)
10     erg = [mod(n,q), erg];
11     n = flor(n/q);
12 end
13
14 z = erg
```

Finden Sie alle **fünf** Fehler und geben Sie an, wie diese ausgebessert werden müssen!
Hinweis: Zeile 2, 8 und 13 sind wegen der Übersichtlichkeit leer!

Lösung:

Die Fehler befinden sich in:

Zeile 1: Natürlich muss die gewünschte Basis q eingegeben werden.

Zeile 3: Ein $=$ bei der Gleichheitsabfrage fehlt.

Zeile 9: Es muss eine Ungleichheitsabfrage sein. Also: “while ($n \neq 0$)”.

Zeile 11: “floor“ schreibt man mit Doppel-o.

Zeile 14: Das $;$ fehlt.

Aufgabe 4:

1. v enthält die echten Teiler der Zahl p .
2. “ p ist eine Primzahl.“
3. Der größte, echte Teiler kann maximal $p/2$ sein. Also braucht die for-Schleife auch nur bis $p/2$ gehen, was die Laufzeit etwa halbiert.

Aufgabe 5:

1. In der letzten for-Schleife müssen die Einträge von $R(k,j) = 0$ sein.
2. Nach der Zeile $[m,n] = \text{size}(A)$ muss eine if-Abfrage eingebaut werden, welche in der vorletzten Zeile durch ein *end* beendet wird:

```
if m == n
    error('A ist nicht quadratisch!')
else
    :
end
```

3. $L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{5}{3} & 1 & 0 \\ \frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix}$ und $R = \begin{pmatrix} 3 & 4 & 2 \\ 0 & \frac{4}{3} & \frac{5}{3} \\ 0 & 0 & \frac{3}{2} \end{pmatrix}$