Praktikum: "Mathematik am Computer" HS 2023
Prof. H. Harbrecht
M. Fallahpour, E. Gajendran, G. Ohlmann, M. Ramm Universität Basel

---

Serie 8

Matlab - Kontrollstrukturen, Repetition

zur 46. KW (13.11. – 17.11.2023)

Aufgabe 8.1 (2 Punkte): In dieser Aufgabe sehen wir, dass es für eine Problemstellung ganz verschiedene Lösungen geben kann. Dazu wollen wir uns anschauen, wie man alle geraden Zahlen von 1 bis 100 aufsummieren kann.

a) Verwende die Gaußsche Summenformel, d.h. es gilt:

$$\sum_{k=1}^{n} 2k = n(n+1).$$

Hinweis: Um alle geraden Zahlen von 1 bis 100 aufzusummieren, muss n in der Gaußschen Summenformel richtig gewählt werden und ist nicht 100.

b) Verwende eine `for`-Schleife.

c) Verwende eine `while`-Schleife.

d) Verwende den Matlab-Befehl `sum`.

e) Verwende eine geschickte Multiplikation mit einem `ones`-Vektor.

f) Du darfst dir gerne noch weitere Möglichkeiten überlegen.

Aufgabe 8.2 (2 Punkte): Schreibe ein Programm, das mit einem Intervall $[a, b]$ startet und dieses *solange* halbiert, bis die Länge kleiner als eine angegebene Toleranz ist. Unterscheide hierbei zwei Fälle:

a) Ersetze jeweils den Endpunkt des Intervalls, d.h. wenn $[a, b_{n-1}]$ das alte Intervall ist, ist das neue von der Form $[a, b_n]$ mit

$$b_n = \frac{1}{2}(b_{n-1} + a).$$

Wiederhole dies *solange*, bis $b_n - a < 10^{-3}$ ist. Setze dabei als Startwerte $a = 0$ und $b = 1$. Zähle auch die Anzahl der Schritte, die dein Programm benötigt hat, und gebe diese aus.

b) Auf die gleiche Weise ersetze nun den Anfangspunkt des Intervalls, d.h. für jedes alte Intervall $[a_{n-1}, b]$ ist das neue von der Form $[a_n, b]$ mit

$$a_n = \frac{1}{2}(a_{n-1} + b).$$

Wiederhole dies *solange*, bis $b - a_n < 10^{-3}$ ist. Setze auch hier die Startwerte $a = 0$ und $b = 1$ und gib die Anzahl der benötigten Schritte aus.

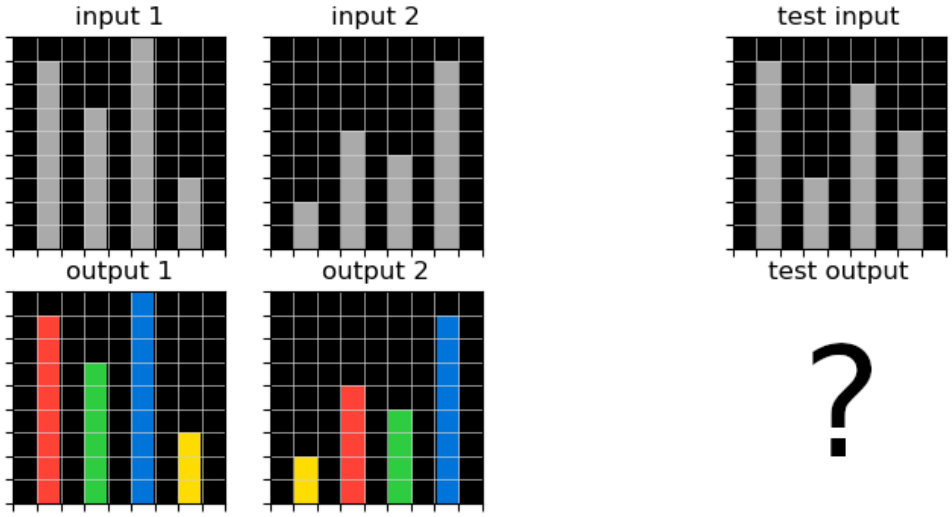
Aufgabe 8.3 (3 Punkte): Betrachte folgendes Glücksspiel:

1. Zu Beginn zahlst du dem Veranstalter einmal einen Einsatz von 10 Franken.
2. Der Veranstalter wirft nun so lange mit einer fairen Münze, bis zum ersten Mal Kopf erscheint. Sei k die Anzahl der dazu notwendigen Versuche.
3. Der Veranstalter zahlt dir nun 2^{k-1} Franken aus.

a) Simuliere dieses Spiel in einem MATLAB-Programm. Du kannst dabei den Befehl `randi` und Schleifenkonstrukte verwenden, um den Zufallsaspekt des Spiels zu implementieren und die Anzahl der Würfe zu zählen. Das Programm soll die Auszahlung und den Gewinn oder Verlust für ein einzelnes Spiel berechnen und ausgeben.

b) Nun wiederhole das Spiel 10^7 mal (`for`-Schleife). Zeichne sowohl die ausbezahlte Summe pro Spiel als auch den Gesamtgewinn in eine Abbildung. Dabei entspricht der Gesamtgewinn vom i-ten Spiel dem addierten Gewinn der Spiele 1 bis i. Was siehst du?

Aufgabe 8.4 (3 Punkte): Betrachte folgende Abbildungen:



In MATLAB können die Abbildungen als 9×9-Matrizen repräsentiert werden, wobei unterschiedliche Farben bestimmten Werten entsprechen:

- Input-Bilder ("input 1", "input 2" und "test input"): Schwarz = 1, Grau = 2.
- Output-Bilder ("output 1" und "output 2"): Schwarz = 1, Gelb = 3, Grün = 4, Rot = 5, Blau = 6.

Die Aufgabe besteht darin, eine MATLAB-Funktion zu erstellen, die eine Input-Matrix in die zugehörige Output-Matrix umwandelt.

a) Erstelle in MATLAB drei Eingabe-Matrizen mit den Namen `I1`, `I2` und `Itest`, die den Bildern "input 1", "input 2" und "test input" entsprechen. Erstelle ebenso zwei Ausgabe-Matrizen `O1` und `O2`, die den Bildern "output 1" und "output 2" entsprechen.

b) Leite die spezifische Transformationsregel ab, die 2-Einträge (graue Kästchen) in den Matrizen `I1` und `I2` in Werte zwischen 3 und 6 in den Matrizen `O1` und `O2` umwandelt. Diese Regel soll aus den zwei gegebenen Paaren von Input- und Output-Bildern abgeleitet werden. Schreibe diese Regel in Wörtern auf. Erstelle dieser Regel folgend die 9×9-Matrix `Otest`.

c) Betrachte die MATLAB-Funktion `O = transformI2O(I)`, die auf der Praktikumswebseite zur Verfügung steht. Setzt diese Funktion deine in b) hergeleitete Transformationsregel um? Ergänze Kommentare im Quellcode der Funktion, um zu erläutern:
   - die Vorgänge innerhalb der `while`-Schleife und der darin verschachtelten `for`-Schleife,
   - den Einsatz und die Funktionsweise der MATLAB-Funktion `sort`, und
   - das Geschehen in der `for`-Schleife danach.

   Hinweis: Am Schluss wird die MATLAB-Funktion `imshow` verwendet, um das zur resultierenden 9×9-Matrix passende Bild anzuzeigen.

Allgemeine Informationen zum Praktikum befinden sich auf der Webseite
`http://cm.dmi.unibas.ch/teaching/praktikumI/praktikumI.html`