Aufgabe 1:

Die Funktion $\mathbf{e^x}$ ist wie folgt über eine unendliche Reihe definiert:

$$e^x = 1 + x/1! + x^2/2! + x^3/3! + x^4/4! + \ \ldots \ = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Formulieren Sie einen Algorithmus in Form eines Flußdiagrammes, der die Berechnung der Summe abbricht, wenn zwei aufeinanderfolgende Summen eine Differenz kleiner als ε haben. Die Varaible ***x*** und ε (z.B. 10^{-8}) sind einzulesen. Der genäherte Wert für $\mathbf{e^x}$ und das erreichte **n** sind auszugeben.

Aufgabe 2:

Die folgenden Summen sind solange zu berechnen, bis der Absolutbetrag des Quotienten eines Summanden und der bisherigen Summe kleiner einer vorher eingelesenen positiven Schranke Epsilon ist. Die Algorithmen sollen als Struktogramme bzw. Flußdiagramme bzw. Programmablaufpläne formuliert werden, optional können C-Programme formuliert werden.

a.) Berechnung von π nach Leibniz: $\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots = \frac{\pi}{4}$

b.) Berechnung von π nach Bailey-Borwein-Plouffe (1996): $\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$

c.) Berechnung von **ln(x)**: $\ln x = \sum_{k=0}^{\infty} \frac{2}{2k+1} \cdot \left(\frac{x-1}{x+1}\right)^{2k+1} \qquad x > 0$

d.) Berechnung von **cos(x)**: $\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} \mp \cdots$

e.) Berechnung von **arctan(x)** für |x| <= 1: $\arctan x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{2k+1} = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \cdots$

Aufgabe 3: Auf Produkte **x, y** und **z** gibt es bei einem Händler folgende Rabatte:

Beim Kauf von Produkt **x** gilt:
- Bis zum 9.Stck. gibt es **keinen** Rabatt, für das 10. - 19.Stück gibt es **10%** Rabatt pro Stück.
- Für das 20. - 39.Stück gibt es **15%** Rabatt pro Stück.
- Für das 40. - 49.Stück gibt es nur noch **5 %** Rabatt pro Stück.
- Für das 50.Stück und weitere Stücke des Produktes x gibt es **keinen** Rabatt.

Beim Kauf von Produkt **y** gilt:
- Ab 6. Stück gibt es **5,- €** Rabatt pro Stück für alle Stücke von 1 an.
- Ab 10. Stück gibt es **10,- €** Rabatt pro Stück für alle Stücke von 1 an.
- Der Gesamtrabatt für Produkt **y** ist auf maximal 140 € beschränkt.

Beim Kauf von Produkt **z** gilt:
- Auf jedes Stück gibt es **8%** Rabatt, ab 25. Stück gibt es **keinen** Rabatt.

Für die drei Produkte **x**, **y** und **z** gemeinsam ist die Gesamtrabattsumme auf **500,- € beschränkt**, d.h. beim Überschreiten von 500,- € Gesamtrabattsumme für einen Warenkorb aus x,y und z gibt es **keinen Rabatt** mehr !

Es ist ein **Flußdiagramm** zu formulieren, welches die **Anzahl** der gekauften Produkte von **x**, **y** und **z** und die Preise/Produkt von **x**, **y** und **z** jeweils getrennt einliest und aufgrund der genannten Vorgaben algorithmisch den **Gesamtpreis** sowohl **ohne** als auch **mit Rabatt** ermittelt und ausgibt. Der **Geldbetrag** des **erzielten Rabatts** ist ebenfalls auszugeben. Beim Einlesen soll sichergestellt werden, daß **Anzahl** ≥ **0** und **Preis** > **0,- €** ist.

Optional kann der Algorithmus als ein C-Programm geschrieben werden.

Aufgabe 4:

(a) Führen Sie das Siebverfahren des Eratosthenes manuell auf Papier für die natürlichen Zahlen von 1 bis 60 durch.

(b) Formulieren Sie einen Programmablaufplan bzw. ein Flußdiagramm bzw. ein Struktogramm, welches das Siebverfahren von Eratosthenes bis n (n > 0) realisiert.

(c) Führen Sie das optimierte Verfahren, ***Algorithmen_Heron_Euklid_Prim.pdf***, ***S. 11***, ***Vorlesung 4*** manuell auf Papier für die natürlichen Zahlen von 1 bis 60 durch.

(d) Formulieren Sie einen Programmablaufplan bzw. ein Flußdiagramm bzw. ein Struktogramm, welches das optimierte Verfahren bis n (n > 0) realisiert.

(e) optional: Schreiben Sie ein C-Programm zum Sieb- und zum optimierten Verfahren

Aufgabe 5: Man teste den rekursiven Algorithmus zur Bestimmung der Primfaktoren am Beispiel ***PFR(330)***. Es ist ein Programmablaufplan bzw. ein Flußdiagramm bzw. ein Struktogramm zu formulieren, welches das rekursive Verfahren beschreibt.

Aufgabe 6: Man teste den iterativen Algorithmus zur Bestimmung der Primfaktoren am Beispiel **`primefactorization(330)`**. Es ist ein Programmablaufplan bzw. ein Flußdiagramm bzw. ein Struktogramm zu formulieren, welches das iterative Verfahren beschreibt.

Aufgabe 7: Man formuliere Programmablaufpläne für eine einfach verkettete Liste mit

struct el { int value; /* Daten */
struct el *next; /* Zeiger auf naechstes Element */
}
struct el *start = 0, *p = 0; /* start ist Startzeiger, p Hilfszeiger */ und gegebenem ***p*** mit

p = (struct el *)malloc(sizeof(struct el)); /* Speicherplatz p auf dem heap */
für

Insertfirst(inout: start, in: p) /* Einfügen p nach start */
Insertlast(inout: start, in: p) /* Einfügen p nach start */
Removefirst(inout: start) /* Löschen erstes Element nach start */
Removelast(inout: start) /* Löschen letztes Element nach start */
int Anzahl(inout: start) /* Rückgabe Anzahl der Listenelement */
Show(in: start) /* Anzeige aller value-Werte */
RemovePos(inout: start, in: pos) /* Element an der Stelle pos löschen (pos = 0 ..) */
Removeall(inout: start) /* Alle Elemente löschen */

Aufgabe 8: Gesucht ist ein Algorithmus in Form eines Flußdiagrammes, Programmablaufplanes oder Struktogrammes zur Berechnung von

$$\sin(x) = \sum_{i=0}^{\infty} (-1)^i \cdot \frac{x^{2 \cdot i + 1}}{(2 \cdot i + 1)!}$$

mit (**0 < x <= π/2**) . Die Berechnung ist abzubrechen, wenn der absolute Betrag des Quotienten aus dem ersten vernachlässigten Reihenglied und der Partialsumme kleiner als der Wert einer positiven Schranke **epsilon** ist. Formulieren Sie den Algorithmus optional als C-Programm.

Aufgabe 9: Es ist je ein Flußdiagramm für folgende Aufgaben zu formulieren, wobei die Vektoren, Matrizen und deren Grenzen als Parameter gegeben sind, die Zulässigkeit der mathematischen Operation (korrekte Indexe) ist jedoch zu überprüfen:
Skalarprodukt zweier Vektoren, ***Summe zweier Vektoren***, ***Multiplikation Matrix mit Vektor***, ***Multiplikation Matrix mit Matrix***, ***Transponieren einer quadratischen Matrix***.

Aufgabe 10: Adi, Bert und Klaus sind beim Einbruch in einen Computerladen erwischt worden. Im Polizeipräsidium geben sie ihre Aussagen zu Protokoll. Kommissar Derrix und sein Assistent Fritz versuchen anhand der drei Aussagen die Tat zu rekonstruieren. Dabei fällt ihnen auf, daß einer gelogen hat. Kommissar Derrix zieht folgende Schlüsse:

1. "Wenn Adi oder Bert die Wahrheit sagen, dann lügt Klaus“
2. "Wenn Bert lügt, dann sprechen entweder Adi oder Klaus die Wahrheit"
3. "Wenn entweder Bert oder Klaus die Wahrheit sagen, dann kann auch Adi nicht lügen"

Sag mir, Fritz, weißt Du jetzt wer gelogen hat ?

Hinweis: Formulieren Sie die Aussagen 1. - 3. als logische Ausdrücke und entwerfen Sie einen Algorithmus als Struktogramm, der alle Kombinationen von "Lügens" und "Wahrheit sagen" für alle 3 Personen mit "brute force" durchläuft. Optional kann der Algorithmus in C implementiert werden.

Wie verändert sich die Lösung, wenn in der 1. Aussage "oder" durch "und" ersetzt wird ?
Wie verändert sich die Lösung, wenn mehrere lügen können ?