

Übung – Grundlagen der Informatik

Zusatzaufgaben - C

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

xckd.com

C: Strukturen, Zeichenketten...

Langweiliges Ratespiel

Schreibe ein Programm, das den Nutzer solange nach einer zufällig gewählten Zahl fragt, bis er sie errät.

Die Zahl soll zwischen 1 und 100 liegen und am Anfang des Programms von der rand()-Funktion bestimmt werden.

(<http://www.cplusplus.com/reference/cstdlib/rand/>)

Der Nutzer soll nach jedem Versuch den Hinweis erhalten, ob seine Zahl zu groß oder zu klein war.

(Bonus: Merke dir die obere und untere Grenze, die bereits vom Nutzer ermittelt wurde und gebe sie vor jedem Rateversuch aus.)

Pythagoras

Finde 3 natürliche Zahlen (a,b,c), die folgende Bedingungen erfüllen:

$$a < b < c$$

$$a^2 + b^2 = c^2$$

$$a + b + c = 1000$$

Palindrom erkennen

Schreibe ein Programm, das prüft, ob ein per Kommandozeilenparameter übergebenes Wort ein Palindrom ist.

(<http://www.c-howto.de/tutorial/funktionen-teil-2/hauptfunktion/>)

Kodierte IP-Adresse

IP-Adressen sind 32 Bit lang und werden deshalb häufig als Integer (int) repräsentiert. Der Mensch liest sie allerdings meistens als vier durch Punkte getrennte 8-Bit-Integer (char). (zB.: 127.0.0.1, 192.168.100.1 etc.)

Schreibe ein Programm, das zum Integer 16777343 die passende, für den Menschen lesbare IP-Adresse ausgibt.

Hinweis: Mit Hilfe eines char-Pointers auf Teile eines int-Werts schauen.

Integral

Schreibe ein Programm, das das bestimmte Integral für $f(x) = x^5 + 2x^2 + x + 1$ zwischen 0 und 2 numerisch berechnet.

Die mathematische Funktion soll mit Hilfe einer separaten C-Funktion implementiert werden, die wiederum von einer „integrate“-Funktion aufgerufen wird. Letzterer soll als Parameter die Intervallsgrenzen für das Integral und eine Schrittweite besitzen.

Bonus: Erweitere die „integrate“-Funktion so, dass sie als ersten Parameter einen Funktionspointer auf eine mathematische Funktion entgegen nimmt.

Taylorreihe der natürlichen Exponentialfunktion

Schreibe ein Programm, das für einen eingegebenen Wert x die natürliche Exponentialfunktion berechnet. Dazu soll eine Taylorreihe verwendet werden. Die Reihenentwicklung soll abbrechen, wenn ein Folgenglied einen Minimalwert unterschreiten ($\varepsilon < 0.0001$).

Achtung: Es sind positive und negative x zulässig!

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad \text{für alle } x \in \mathbb{R}$$

<https://de.wikipedia.org/wiki/Math.h>

Messreihe

Schreibe ein Programm zur Simulation einer Messreihe.

Eine Messung soll als Struktur definiert werden und folgende Felder enthalten:

- ID der Messung (ganzzahlig, positiv)
- Messwert (Gleitkommazahl, doppelte Genauigkeit)
- Zeichenkette, die den Namen des Messgeräts enthält (7 sichtbare Zeichen lang)
- Pointer auf die vorherige Messung

Der Nutzer soll beim Aufrufen des Programms eine Zahl als Kommandozeilenparameter an die main-Funktion übergeben. (<http://www.c-howto.de/tutorial/funktionen-teil-2/hauptfunktion/>)

Es soll eine Messreihe mit dieser Anzahl an Messungen generiert werden. Die IDs sollen fortlaufend verteilt werden. Alle Messungen wurden mit dem gleichen Messgerät erfasst („FX-9000“). Der Wert der Messungen soll mit Hilfe der Zufallsfunktion rand() gewählt werden und zwischen 0 und 5000 liegen. (<http://www.cplusplus.com/reference/cstdlib/rand/>)

Erstelle eine Funktion zur Ausgabe einer Messung. Eine weitere Funktion soll den Mittelwert aller Messwerte berechnen. Gib anschließend alle Messwerte aus, die über dem Mittelwert liegen.

Pong

Schreibe eine Terminalanwendung, die dem bekannten Klassiker „Pong“ entspricht.

Verwende für die Darstellung die Bibliothek *ncurses*.

- *ncurses* muss installiert sein
- `#include<curses.h>`
- Kompilieren mit: `gcc pong.c -lncurses`

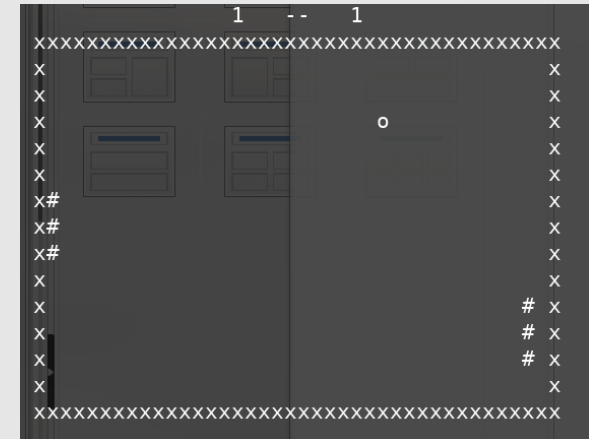
Hinweise:

Lege eine Datenstruktur für Positionen, sowie eine für die Haltung aller

Statusinformationen des Spiels an. (Position Padel links/recht, Position Ball, Bewegungsrichtung Ball, Punktestand, Pause-Flag, Ende-Flag)

Implementiere die Funktionen `key_proc(...)`, `detect_collision(...)`, `move_ball(...)`, `print_field(...)` und rufe sie, zusammen mit `usleep(50000)`, in einer Endlosschleife auf.

Nach dem Erstellen eines „Fensters“ mit der *ncurses*-Funktion `initscr()` folgende Funktionen ausführen: `noecho()` - verhindert die Ausgabe der Tastatureingaben, `cbreak()` - verhindert das Zeilen-Buffern der Nutzereingabe, `keypad(stdscr, TRUE)` - aktiviert Pfeil- und andere Funktionstasten, `nodelay(stdscr, TRUE)` - macht `getch()` nicht-blockierend (mit `getch()` kann die Tasteneingabe eingelesen werden)



Wochentage berechnen

Schreibe ein Programm, das zu einem eingegebenen Datum (dd.mm.yyyy) den Wochentag berechnet.

Dazu kann man von einem bekannten Datum ausgehen und den Abstand zu diesem Datum berechnen. Mit Hilfe des modulo-Operators (%) lässt sich dann ein Wochentag bestimmen.

Der 1.1.1 war z.B. ein Samstag.

Achtung!:

- Schaltjahre beachten, Julianischer und Gregorianischer Kalender (Umstellung 1700 in „Deutschland“, Sep. 1752 in GB)