

- Achten Sie darauf, nicht zu lange Zeilen, Funktionen und Dateien zu erstellen.
- Programmcode muss in englischer Sprache verfasst sein.
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich.
- Die Kommentare sollen einheitlich in englischer Sprache verfasst werden.
- Verwenden Sie keine Bibliotheken ausgenommen die Standard C und Mathematik Library (libc und libm), es sei denn die Aufgabenstellung erlaubt ausdrücklich weitere Bibliotheken.
- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Halten Sie die Regeln zu Variablen-, Funktionen- und Dateibenennungen ein und wählen Sie aussagekräftige Namen.
- Allgemeiner Hinweis: Bei Regelverletzung wird der Praktomat Ihre Abgabe zurückweisen.
- Ausgaben von Fließkommazahlen sollen auf 2 Stellen nach dem Komma begrenzt werden

### Abgabemodalitäten

Die Praktomat-Abgabe wird am 15. Mai 2022 freigeschaltet.

Achten Sie unbedingt darauf, diese Dateien im Praktomat bei der richtigen Aufgabe hochzuladen. Falsch hochgeladene Abgaben werden nicht bewertet. Bitte beachten Sie, dass das erfolgreiche Bestehen der öffentlichen Tests für eine erfolgreiche Abgabe nötig ist. Planen Sie für Ihren ersten Abgabeversuch entsprechend Zeit ein. Bitte beachten Sie: Sollten Sie Probleme bei der Abgabe bzw. beim Hochladen Ihrer Dateien haben, verwenden Sie einen anderen Browser. Der Internet Explorer kann Probleme verursachen. Die Dateien node.h und node.c sind **nicht** abzugeben.

# Jukebox

Ziel: Dateiverarbeitung und Stringverarbeitung

Auf einer Jukebox existiert eine Liste von Titeln (jukebox.txt). Die Aufgabe ist es, mit einem Programm die Suche nach Musiktiteln oder auch Interpreten zu unterstützen.



- Erstellen Sie in dem C-Programm die Funktion **Track\_Search**.
- Track\_Search wird der Such-String und ein Zeiger auf FILE übergeben
- Track\_Search durchsucht in der Datei nach dem Such-String.
- Verwenden Sie in Track\_Search die Funktion **strstr** aus **string.h**

## Eingabe:

Der Benutzer soll durch eine Eingabe die Möglichkeit haben eine Dateiname und einen Suchstring einzugeben

Bitte Dateiname eingeben: ./src/jukebox.txt

Bitte den Suchstring eingeben: Robbie

## Ausgabe:

Ausgegeben sollen alle Treffer aus der jukebox.txt. Folgenden Informationen sollen ausgegeben werden: Track Nummer, Musiker und Musiktitel

Track 13: Robbie Williams – Supreme

Track 14: Robbie Williams – Advertising Space

Track 15: Robbie Williams – Sin Sin Sin

Track 73: Robbie Williams – Feel

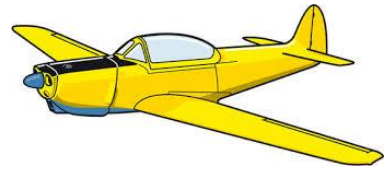
Quelle für die Musiktitel: <https://www.musiksocke.de/2000er-musik>

# Kokosnuss Wings



Ziel: Strukturen, Zeiger und Listen

Stellen Sie sich vor, Sie leiten ein Tourismusunternehmen, das Flugzeugreisen zwischen Inseln organisiert. Jede Tour besteht aus einer Folge kurzer Flüge von einer Insel zur nächsten. Für jede der Inseln müssen Sie einige Daten aufzeichnen, z.B. den Namen der Insel und die Öffnungszeiten des Flughafens. Der Pilot von Kokosnuss Wings soll nun die Möglichkeit bekommen, sich mit einem Programm durch die Eingabe der Tournummer, eine Liste von Inseln mit den jeweiligen Öffnungszeiten ausgeben zu lassen.



Die Daten der Inseln sollen in einer Struktur gehalten werden.

Inseldaten:

Amity von 9:00 bis 17:00 Uhr  
Craggy von 9:00 bis 16:00 Uhr  
Isla Nubar von 7:00 bis 12:00 Uhr  
Shutter von 8:00 bis 20:00 Uhr  
Skull von 8:00 bis 17:30 Uhr

Für Ihr Unternehmen bieten Sie vier Touren an:

Tour1 - Amity, Craggy, Isla Nubar, Amity  
Tour2 - Skull, Craggy, Isla Nubar, Skull  
Tour3 - Shutter, Skull, Shutter  
Tour4 - Isla Nubar, Skull, Shutter, Amity, Isla Nubar

Vorgehen:

- a) Erstellen Sie die Datenstruktur Insel in der Header-Datei „insel.h“. Zusätzlich noch die folgenden Funktionsdeklarationen.

```
void Input_Insel(Insel * theInsel, char* name, Time open, Time closed);  
void Output_Insel(Insel * theInsel);
```

- b) Um die Touren zu verwalten verwenden Sie aus der Vorlesung die Datenstruktur Node. Node soll ein Element/Attribut der Struktur Insel enthalten. Die Dateien „node.c“ und „node.h“ werden zur Verfügung gestellt.
- c) In der Datei „insel.c“ sollen die Funktionen Input und Output Insel implementiert werden.
- d) Erstellen Sie für jede Tour eine separate Liste im Hauptprogramm.  
(Beispiel: `Node * tour1;`)
- e) In der Datei „main.c“ legen Sie die einzelnen Inseln und Touren mit den Funktionen Setup\_Insel und Setup\_Tour an. Mit der Funktion Output\_Tour, die einen Zeiger auf die Liste und die Tour Nummer übergeben bekommt, wird die Information für den Piloten ausgedruckt. In der Funktion main sollen die Setup Funktionen aufgerufen werden und die Tour Nummer abgefragt werden. Mit

einem switch case soll die Output\_Tour Funktion jeweils mit der richtigen Tour und der eingegebenen Tour Nummer aufgerufen werden. Wird eine Tour Nummer 0 eingegeben, soll sich das Programm beenden, in allen anderen Fällen soll das Programm weiter nach einer Tour Nummer fragen.

**Eingabe:**

Bitte die Tour Nummer (1-4) eingeben oder mit 0 beenden: 1

**Ausgabe:**

```
-----  
----- Tour 1 -----  
-----  
Insel          : Amity  
Öffnungszeiten: 09:00 bis 17:00  
-----  
Insel          : Craggy  
Öffnungszeiten: 09:00 bis 16:00  
-----  
Insel          : Isla Nubar  
Öffnungszeiten: 07:00 bis 12:00  
-----  
Insel          : Amity  
Öffnungszeiten: 09:00 bis 17:00  
-----  
                                     Ende
```