



Allgemeine Hinweise

- Achten Sie auf die korrekte Benennung Ihrer Dateien, damit diese vom Praktomaten akzeptiert werden.
- Ausgaben der einzelnen Programme sind exakt nach den Vorgaben der Aufgabenstellung zu realisieren.
- Programmieren Sie auf Englisch:
 - Sämtliche in Ihren Programmen verwendete Variablen und Funktionen müssen auf Englisch sein.
 - Das Programm ist mit so vielen wie nötig, so wenig wie möglich sinnvollen Kommentaren auf Englisch zu beschreiben.
- Die Gesamtaufgabe ist sinnvoll in Unterfunktion aufzuteilen.
 - Unterfunktionen, die in dem jeweiligen C-Programmen Verwendung finden, sollen als Funktionsprototypen vordefiniert und nach der Main oder in einer .c Datei programmiert werden.
 - Die main soll lediglich für Ein- und Ausgabe (Funktionen) sowie die Aufgabe lösende Funktionsaufrufe genutzt werden.
- Vorgegebene Zahlenwerte sollen als symbolische Konstanten definiert und als solchen verwendet werden. Die Verwendung von globalen Variablen führt zu Punktabzug!
- Lesen sie die gesamte Aufgabenstellung vor der Bearbeitung sorgfältig durch. Sie finden für jede Teilaufgabe unterhalb der Aufgabenbeschreibung einen Korrekturschlüssel an dem sich für die Korrektur orientiert wird.
- Außer andersweitig in den Aufgabenstellungen beschrieben sind nur die Bibliotheken `stdio.h`, `stdlib.h`, `math.h` und `string.h` zugelassen. Eine eigene selbstgeschriebene .h-Datei dürfen sie verwenden.



Aufgabe 4-1 Jukebox

Aufgabenbeschreibung:

Auf einer Jukebox existiert eine Liste von Titeln, die in einer Textdatei „jukebox.txt“ gespeichert sind. Die Aufgabe ist es, mit einem Programm die Suche nach Musiktiteln oder auch Interpreten zu unterstützen.

Die Datei `jukebox.txt` kann zum lokalen testen auf Ilias gefunden werden. Beim Hochladen in den Praktomat muss die `jukebox.txt` nicht mit hochgeladen werden.

- Erstellen sie in einem C-Programm die Funktion `Track_Search`.
- `Track_Search` wird ein Zeiger auf `FILE` und der vom User festgelegte Suchstring übergeben.
- `Track_Search` durchsucht die Datei nach Werten außerhalb der Fehlerschranke.
- Verwenden Sie in `Track_Search` die Funktion
`char *strstr(const char *haystack, constchar *needle)`
aus der String Bibliothek `string.h`.

Zuerst soll der Nutzer nach der Name der Datei und einem Such-Begriff gefragt werden, dann sollen alle lzu dem Suchbegriff passende Tracks ausgegeben werden.

Beispiel

Eingabe:

Die nutzende Person muss zuerst den Namen der Datei und dann in einer zweiten Abfrage den Suchbegriff eingeben:

```
Please enter a Jukebox file name: ./jukebox.txt
```

```
Please enter a search string: Robbie
```

Ausgabe:

Für alle Treffer aus der Datei sollen folgende Informationen ausgegeben werden:

- Track Nummer
- Musiker:in
- Musiktitel

```
Track 13: Robbie Williams - Supreme  
Track 14: Robbie Williams - Advertising Space  
Track 15: Robbie Williams - Sin Sin Sin  
Track 73: Robbie Williams - Feel
```

Zusätzliche Hinweise zur Abgabe 4-1

Korrekturschlüssel Abgabe 4-1

Insgesamt können mit der Aufgabe 15 Punkte erreicht werden. Die Verteilung kann wie folgt angenommen werden:



Programmfunktion:	3 Punkte
Korrekte Ausgabe	3
Leserlichkeit und Qualität	2 Punkte
Codeformat	1
Kommentare	1
Summe	5 Punkte

Aufgabe 4-2 Coconut Wings

Aufgabenbeschreibung

Das Tourismusunternehmen „Coconut-Wings“ bietet Flugreisen zwischen beliebten Inseln an. Jede dieser Touren besteht aus einer Folge kurzer Flüge von Insel zu Insel. Zur besseren Planung dieser Touren sollen Sie nun ein Programm entwickeln, womit ein Nutzer, also die Dispatcher und Piloten, durch Eingabe einer Tournummer die Öffnungszeiten der jeweiligen Inseln ausgegeben bekommen kann.

- Überlege dir Strukturen mit denen du sinnvoll die Daten der Inseln speichern kannst
- Schreibe eine Funktion die deine Struktur mit Daten befüllt und für die Touren jeweils eine doppelt verkettete Liste erstellt.
- Frage den Nutzer nach einer Tournummer und drucke die entsprechenden Informationen wie im Beispiel zu sehen in die Konsole.
- „Tour“-Nummer „0“ soll das Programm beenden ohne etwas auszugeben.

Daten:

Die Flüge von „Coconut-Wings“ fliegen die folgenden Inseln an:

- **Amity Island:** Idyllische Urlaubsinsel, die besonders durch den Artenreichtum in den angrenzenden Gewässern, besonders die großen Haivorkommen sind hervorzuheben.
- **Craggy Island:** Diese abgelegene, ruhige, grüne Insel ist besonders bekannt für ihre malerisch gelegene Parochialkirche.
- **Isla Nubar:** Einmaliges Naturdenkmal, welches weltweite Berühmtheit erlangte durch den dort angesiedelten Freizeitpark, in welchem weltweit einzigartige Reptilien bewundert werden können.
- **Shutter Island:** Dieser Ausflug ist nichts für schwache Nerven! Besonders bei Lost-Place-Begeisterten ist die inzwischen verfallene Nervenheilanstalt auf der Insel ein begehrtes Ziel für Erkundungen.
- **Skull Island:** Die tropische Insel mit atemberaubender Geographie ist die Heimat einzigartiger, teils ausgestorbener geglaubter Tierarten. Zu besonderer Berühmtheit gelangte sie durch die Entdeckung einer einzigartigen Menschenaffenart.

Die Inseln haben folgende Öffnungszeiten:

- **Amity Island:** 9:00 bis 17:00 Uhr
- **Craggy Island:** 9:00 bis 16:00 Uhr
- **Isla Nubar:** 7:00 bis 12:00 Uhr



- **Shutter Island:** 8:00 bis 20:00 Uhr
- **Skull Island:** 8:00 bis 17:30 Uhr

Es werden vier verschiedene Touren angeboten:

- **Tour 1:** Amity - Craggy - Isla Nubar - Amity
- **Tour 2:** Skull - Craggy - Isla Nubar - Skull
- **Tour 3:** Shutter - Skull - Shutter
- **Tour 4:** Isla Nubar - Skull - Shutter - Amity - Isla Nubar

Beispiel

Eingabe:

Die nutzende Person muss die gewünscht Tour-Nummer eingeben:

```
Please enter a tour number (1-4) or enter the number 0 for exit the program: 1
```

Ausgabe:

Die entsprechende Tour soll ausgegeben werden oder das Programm beendet werden:

```
-----  
----- Tour 1 -----  
-----  
Island : Amity  
Opening hours: 09:00 - 17:00  
-----  
Island : Craggy  
Opening hours: 09:00 - 16:00  
-----  
Island : Isla Nubar  
Opening hours: 07:00 - 12:00  
-----  
Island : Amity  
Opening hours: 09:00 - 17:00  
-----  
End
```

Zusätzliche Hinweise zur Abgabe 4-1

Korrekturschlüssel Abgabe 4-1

Insgesamt können mit der Aufgabe 15 Punkte erreicht werden. Die Verteilung kann wie folgt angenommen werden:



Programmfunktion:	7 Punkte
Korrekte Ausgabe	5
Sinnvolle Struktur zum Speicher der Daten	1
„Tour“ 0 beendet das Programm	1
Leserlichkeit und Qualität	3 Punkte
Codeformat	2
Kommentare	1
Summe	10 Punkte

Aufgabe 4-3 Shuttle Sortierung

Aufgabenbeschreibung:

Bei dieser Aufgabe soll eine Liste mit den Starts des amerikanischen Space Shuttles entsprechend der Missionsbezeichnungen alphabetisch absteigend (A vor Z) sortiert werden.

Die Liste soll dabei aus einer `shuttle_data.txt` Datei ausgelesen werden die für lokale Tests in Ilias gefunden werden kann. Die sortierte Liste soll dann in vergleichbarer Formatierung wie in der `.txt`-Datei in die Konsole ausgegeben werden. Die Formatierung kann dem Beispiel weiter unten entnommen werden. Beim hochladen in den Praktomaten ist es nicht notwendig `shuttle_data.txt` mit hochzuladen.

- Erstelle eine Funktion um die `.txt`-Datei einzulesen und in einer (doppelt) verketteten Liste zu speichern.
- Implementiere eine Sortieralgorithmus deiner Wahl um die Liste zu sortieren.
- Erstelle eine Funktion um die sortierte Liste in vergleichbarer Formatierung in die Konsole auszugeben.

Beispiel:

Auszug der `shuttle_data.txt`-Datei:

```
Missions-Nr. | Start-Datum | Missionsbezeichnung | Orbiter | Startrampe
-----
1 | 12.04.1981 | STS-1 | Columbia | LC-39A
34 | 28.02.1990 | STS-36 | Atlantis | LC-39A
98 | 19.05.2000 | STS-101 | Atlantis | LC-39A
```

Beispielhafte Konsolen Ausgabe des Auszuges:

```
Missions-Nr. | Start-Datum | Missionsbezeichnung | Orbiter | Startrampe
-----
1 | 12.04.1981 | STS-1 | Columbia | LC-39A
98 | 19.05.2000 | STS-101 | Atlantis | LC-39A
34 | 28.02.1990 | STS-36 | Atlantis | LC-39A
```



Zusätzliche Hinweise zur Abgabe 4-3

Korrekturschlüssel Abgabe 4-3

Insgesamt können mit der Aufgabe 15 Punkte erreicht werden. Die Verteilung kann wie folgt angenommen werden:

Programmfunktion:	10 Punkte
Korrekte Ausgabe	7
Korrektes Einlesen der .txt Datei	1
Verwenden einer (doppelt) verketteten Liste	1
Implementierung eines Sortieralgorithmus	1
Leserlichkeit und Qualität	5 Punkte
Codeformat	3
Kommentare	2
Summe	15 Punkte