

Aufgabe 1. ▲▲

In der Casino-Welt gehört Blackjack zu den beliebtesten Tischspielen. In dieser Aufgabe soll zunächst eine vereinfachte Version des Spiels in C implementiert werden. Das (hier vereinfachte) Spiel funktioniert wie folgt: Es steht ein Kartendeck mit den Karten von 2 bis 10 bereit. Die Wertigkeit der Karten gleicht ihrer Zahl. Ein einzelner Spieler tritt direkt gegen das Haus an. Das Spiel lässt sich in Phasen gliedern:

- Anfangs zieht das Dealer eine Karte vom Stapel und deckt diese auf.
- Anschließend kann der Spieler so viele Karten ziehen, wie er möchte. Sein Ziel sollte es sein, mit der Summe seiner Kartenwerte möglichst nah an 21 Punkte zu kommen, ohne sie zu überschreiten. Überschreitet er in der Summe die 21, hat er sofort verloren. Liegt er darunter und möchte keine weitere Karte ziehen ("stay"), beginnt die nächste Phase:
- Der Dealer deckt so lange Karten auf, bis er inklusive der zuerst gezogenen Karte in der Summe 16 Punkte überschreitet. Danach darf er keine weitere Karte mehr aufdecken. Hat er dabei die 21 Punkte-Grenze überschritten oder hat er weniger Punkte als der Spieler, verliert er. Ansonsten gewinnt das Haus.

Implementation von Black Jack

(Zu der Aufgabe gibt es eine vorgefertigte C-Datei die ihr verwenden könnt, aber nicht müsst)

- a) Implementieren Sie die Funktion "int karteziehen()", die die Funktion rand verwendet, um eine Zufallszahl zwischen 2 und 10 zu generieren und diese zurückgibt. **Tipp:** Nutzen Sie den Modulo-Operator % , um die Zahlen in ihrer Größe zu begrenzen. ▲▲
- b) Schreiben Sie anschließend ein Programm, welches die oben beschriebene Form von Blackjack implementiert ▲▲

Aufgabe 2. ▲ – ▲▲▲

Schreibe ein Modul `arrayhelpers`, das einige nützliche Funktion zum Umgang mit `int`-Arrays enthält:

- a) Array zeilenweise oder mit Kommata getrennt ausgeben ▲

- b) Alle Felder eines Arrays mit einem Wert initialisieren ▲
- c) Array um 1 rotieren (d.h. das hinterste Element an erste Stelle schreiben und alle anderen Elemente um eins nach hinten schieben) ▲▲
- d) Array um k rotieren ▲▲
- e) Array umdrehen ▲▲
- f) Ein Array in einem anderen suchen und die Position zurück geben. Sollte das Array nicht im anderen enthalten sein, so soll der Rückgabewert -1 sein. ▲▲▲

Beispiel:

```
1  int A[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
2  int B[3]  = {4, 5, 6}  
3  int C[2]  = {5, 7}  
4  int D[2]  = {9, 10}
```

Hier gilt: B ist an 3-ter Stelle in A enthalten und D an 8-ter. Das Array C ist garnicht in A enthalten, darum wird der Rückgabewert -1 sein.

Aufgabe 3. Umgang mit Strings (Teil 1): ▲

Palindrome sind Wörter, die vorwärts und rückwärts gelesen gleich sind, z.B. Ebbe.

- a) Schreiben Sie eine Funktion, die als Argument einen String erhält und diesen umgekehrt ausgibt
- b) Schreiben Sie eine Funktion, die zurückgibt, ob ein als Argument übergebender String ein Palindrom ist.