

### Beispiel1: Kartenspiel (20 Punkte)

Sie haben ein Kartenspiel mit 52 Karten. Diese initialisieren Sie.

Danach 10 mal:

Karten mischen

Jeweils 7 Karten an 4 Mitspieler ausgeben

Prüfung, wieviele „Flush“ ausgegeben wurden.

Am Ende geben Sie das Ergebnis aus, d.h. wieviele Flush hatte jeder Spieler (Flush=mind. 5 Karten einer Farbe).

```
enum farbe {herz, karo, kreuz, pique};
enum wert {zwei, drei, vier, fuenf, sechs, sieben, acht, neun,
zehn, bube, dame, koenig, as};
struct karte {
    farbe f;
    wert w;
};
```

```
const int anz_spieler = 4;
const int hand_groesse = 7;
typedef karte stapel[52];
typedef karte gegner[anz_spieler][hand_groesse];
typedef int ergebnisse[anz_spieler];
```

```
void initialisieren(stapel st, ergebnisse e);
void mischen(stapel st);
void verteilen(stapel st, gegner g);
void bewerten(gegner g, ergebnisse e);
void ausgabe_ergebnisse(ergebnisse e);
void drucke_karten(gegner g);
```

Implementieren Sie die hier angegebenen Funktionen und zusätzlich die Funktion „Flush“, überlegen Sie dafür eigenständig, wie deren Schnittstelle auszusehen hat.

### Beispiel2: Überladene Funktionen (16 Punkte)

Schreiben Sie für die Typen „unsigned“, „int“, „short“ und „char“ eine Funktion mit der Bezeichnung „bitausgabe“, die für das übergebene Argument die Binärrepräsentation am Bildschirm ausgibt.

```
int i = -1;
char c = 7;

bitausgabe(c);           // 00000111
bitausgabe(i);          // 11111111111111111111111111111111
```

**Abgabe:** bis 20.11. 23:59 Uhr elektronisch (Verzeichnis uebung2 einrichten!) und am 21.11. zu Beginn des Praktikums auf Papier (Listing der Quelldateien).