

# Grundlagen der Informatik II – 5. Übung

Schwerpunkt: Komplexbeispiel zur Anwendung dynamischer Speicherverwaltung

## Aufgabenstellung

Es soll eine Reihe von Funktionen zur Verwaltung eines dynamischen Float-Vektors erstellt werden. Dieser soll sich z.B. automatisch vergrössern, wenn neue Werte eingefügt werden. Zu Beginn soll eine bestimmte Anzahl Elemente vorreserviert werden. Wenn diese überschritten wird, soll der Vektor vergrössert werden. Dabei ist stets die maximale Feldgrösse und die davon benutzte Grösse zu protokollieren.

## Datenstruktur „FloatVec“

```
struct FloatVec {  
    float *vecPtr; // Zeiger auf ein float-Feld  
    int maxmem;    // max. verfügbare Elemente  
    int used;      // belegte Elemente  
};
```

## Funktionen zur Verwaltung

- void fv\_init(FloatVec& v, int maxmem=10)  
*Vektor mit einer bestimmten Grösse initialisieren (Standard: 10)*
- void fv\_resize(FloatVec& v, int size)  
*Vektorspeicher auf eine neue Grösse vergrössern*
- void fv\_append(FloatVec& v, float wert)  
*Wert an den Vektor anhängen*
- void fv\_print(const FloatVec& v, int precision=2)  
*Float-Vektor ausgeben mit gewünschter Stellenanzahl nach Komma*
- void fv\_delete(FloatVec& v)  
*Vektor leeren*

## Anwendung der Float-Vektor-Funktionen

```
FloatVec fv;  
fv_init(fv,5);  
for(float i=0.1; i<2.0; i+=0.4)  
    fv_append(fv,i);  
fv_print(fv,5);  
fv_append(fv,123.45);  
fv_print(fv,2);  
fv_delete(fv);
```