

## Blatt 3

### Aufgabe 1 (Pointer verstehen; Votieraufgabe)

In den nachfolgenden Codeteilen werden sinnfrei Pointer verwendet. Erklären Sie was in den jeweiligen Zeilen und Schleifendurchläufen passiert:

Teil 1:

```
int a=1, b=10, c =100;
int *p1=&a, *p2=&b, *p3=&c;

*p1 = *p1 + *p2 + *p3;
*p2 = *p3 - *p2 - *p1;
*p3 = 1** p1 + 2** p2 + 3** p3;
```

Teil 2:

```
int d=1, e=2, f=3, *p1=&d, *p2=&e, *p3=&f;

*p1 = *p1 ** p2 ** p3 ;
*p3 = (* p1)++ + ++(*p2);
*p3 *= *p2 ;
```

Teil 3:

```
int g=1, h=6, i=7, *p1=&g, *p2=&h, *p3=&i;

for (int ind =*p1 ; ind <=*p3 ; ind++){
(* p1)++;
(* p2)+=2;
--(*p3);
}
```

Teil 4:

```
int j=1, k=10, l=0, *p1=&j, *p2=&k, *p3=&l;

for (int *ind=p2 ; *ind >= p1 ; *ind --*p1){
(* p3 ) += (* ind ) * (* ind ) ;
}
```

### Aufgabe 2 (Kahan; Votieraufgabe)

Vergleichen Sie die Kahan-Korrektur für das Skalarprodukt hinsichtlich Geschwindigkeit und Genauigkeit mit einer naiven Implementierung.

### Aufgabe 3 (Methoden; Votieraufgabe)

Schreiben Sie eine Funktion

- die als Argument zwei Vektoren **a** und **b**, deren Länge **n** und einen Integer **method** akzeptiert,
- wobei **method** zwischen drei verschiedenen Implementierungen aus dem Skalarprodukt auswählt (bspw. Standard, Kahan und Vorsortiert)

- und jede Implementierung in einer eigenen Datei definiert wird.

*Hinweis:*

1. *Hierfür müssen Sie zunächst die jeweiligen C-Dateien zu Objekten \*.o kompilieren, welche dann zu einer Binary gelinkt werden.*
2. *Um eine Funktion zu verwenden, die außerhalb einer Datei definiert wird, muss sie innerhalb der ausführenden Datei deklariert werden, bspw. in der Form:*

```
double scalar1(double *a, double *b, int n, int method);
```