



5. Übung zur Vorlesung Programmierung im Sommersemester 2019

Abgabe: Mittwoch, den 22.05.2019 bis spätestens 12 Uhr.

Aufgabe 5.1. (6 Punkte) Funktionen und Prozeduren

Funktionen dienen dazu, häufig genutzte Programmteile auszulagern und zu generalisieren. Dadurch werden Programme besser lesbar und sind einfacher zu warten/debuggen. Des Weiteren wird der Programmieraufwand geringer. Im Folgenden sollen einige nützliche Funktionen und Prozeduren für integer Arrays implementiert werden.

- (a) Implementieren Sie eine Prozedur mit dem Prototyp

```
void print_int_array(int a[], size_t length);
```

welche ein integer Array `a` und dessen Länge `length` als Argumente erhält. Die Prozedur soll nun in einer ersten Zeile die Arrayindizes und in der zweiten Zeile die zugehörigen Werte des Arrays ausgeben. Testen Sie die Prozedur, indem Sie ein Array initialisieren und es ausgeben. Vergessen Sie nicht, dass die Indizierung bei 0 beginnt!

- (b) Implementieren Sie folgende Funktionen

```
int get_min(int a[], size_t length);  
int get_max(int a[], size_t length);  
double get_mean(int a[], size_t length);  
double get_variance(int a[], size_t length);
```

die den kleinsten Wert, den größten Wert, das arithmetische Mittel

$$\bar{a} = \frac{1}{\text{length}} \sum_{i=0}^{\text{length}-1} a_i, \quad (1)$$

und die Varianz

$$\text{Var}(a) = \frac{1}{\text{length}} \sum_{i=0}^{\text{length}-1} (a_i - \bar{a})^2, \quad (2)$$

der Werte des Arrays `a` zurückgeben sollen. Testen Sie erneut die Funktionen mit einem Array Ihrer Wahl.

- (c) Schreiben Sie eine Funktion

```
size_t find_element(int val, int a[], size_t length);
```

welche den kleinsten Arrayindex ausgibt, an dem das Element von `a` den Wert `val` hat. Falls kein Element gefunden wird, soll `length` zurückgegeben werden.

Aufgabe 5.2. (4 Punkte) Rekursion

Diese Aufgabe dient dazu rekursive Funktionen zu definieren und anzuwenden.

- (a) Schreiben Sie eine Funktion `'size_t rec_factorial(size_t n)'`, welche die Fakultät einer Zahl $n > 0$ mit Hilfe von Rekursion berechnet.
- (b) Implementieren Sie daraufhin eine Prozedur

```
void print_fibonacci(unsigned int a0, unsigned int a1, size_t n);
```

die zwei positive natürliche Zahlen $a_0, a_1 \in \mathbb{N}$ sowie $n \in \mathbb{N}$ als Argumente erhält und daraufhin die durch

$$a_{j+2} = a_{j+1} + a_j \tag{3}$$

definierte Folge bis $j = n$ ausgibt.

Aufgabe 5.3. (4 Punkte) Sort and find

Implementieren Sie einen Sortieralgorithmus, sowie einen effizienten Suchalgorithmus!

- (a) Schreiben Sie eine Prozedur

```
void sort(int a[], size_t length);
```

welche ein integer Array `a` der Länge `length` einliest und die Einträge umsortiert, so dass nach Ausführung der Prozedur die Werte von `a` aufsteigend geordnet sind, d.h. $a_i \leq a_j$ für $1 \leq i < j \leq \text{length}$. Sie dürfen `get_max` und `find_element` aus Aufgabe 1 verwenden.

- (b) Wir wollen jetzt einen effizienten Suchalgorithmus implementieren, der auf aufsteigend sortierte Arrays anwendbar ist. Schreiben Sie hierzu die Funktion

```
size_t find(int var, int a[], size_t length);
```

die ein mittels `sort` vorsortiertes Array erhält und in diesem Array den Wert `var` sucht. Falls erfolgreich wird der zugehörige Index zurückgegeben, ansonsten `length`. Gehen Sie wie folgt vor:
Schreiben Sie eine Funktion

```
size_t find_in_range(int a[], size_t lower, size_t upper);
```

die innerhalb von $\text{lower} \leq i \leq \text{upper}$ nach dem Wert sucht. Die Funktion soll dann den Wert des Intervalls an $(\text{upper} - \text{lower})/2$ mit dem Wert von `val` vergleichen. Falls gleich, ist der Wert gefunden und der Index wird ausgegeben. Falls ungleich wird je nachdem ob der Wert größer oder kleiner war die Funktion rekursiv auf einem der beiden Teilintervalle aufgerufen. Ist die Bedingung `lower == upper` erfüllt und der zugehörige Arraywert immer noch ungleich zu `val`, so ist `val` nicht vorhanden und `length` wird zurückgegeben. Implementieren Sie dann `find` indem Sie `find_in_range` auf das gesamte Array anwenden. Testen Sie die Funktion anhand eines Arrays Ihrer Wahl.