



Programmierblatt 01

Ausgabe: 02.11.2022 18:00

Abgabe: 17.11.2022 20:00

Thema: Pseudocode, Insertionsort

Abgabemodalitäten

1. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf Ihrem Rechner mit dem Befehl `clang -std=c11 -Wall -g` kompilieren.
2. Die Abgabe für den Quellcode erfolgt ausschließlich über unser Git im entsprechenden Branch. Nur wenn ein Ergebnis im [ISIS-Kurs](#) angezeigt wird, ist sichergestellt, dass die Abgabe erfolgt ist. Die Abgabe ist bestanden, wenn Sie an Ihrem Test einen grünen Haken sehen.
3. Sie können bis zur Abgabefrist beliebig oft neue Versionen abgeben. Lesen Sie sich die Hinweise der Tests genau durch, denn diese helfen Ihnen die Abgabe zu korrigieren.
Bitte beachten Sie, dass ausschließlich die letzte Abgabe gewertet wird.
4. Die Abgabe erfolgt, sofern nicht anders angegeben, in folgendem Branch: `iprg-b<xx>-a<yy>`, wobei `<xx>` durch die zweistellige Nummer des Aufgabenblattes und `<yy>` durch die entsprechende Nummer der Aufgabe zu ersetzen sind.
5. Geben Sie für jede Aufgabe die Quelldatei(en) gemäß der Vorgabe ab. Im [ISIS-Kurs](#) werden zum Teil Vorgabedateien bereitgestellt. Nutzen Sie diese zur Lösung der Aufgaben.
6. Die Abgabefristen werden vom Server überwacht. Versuchen Sie Ihre Abgabe so früh wie möglich zu bearbeiten. Damit minimieren Sie auch das Risiko, die Abgabefrist auf Grund von „technischen Schwierigkeiten“ zu versäumen. Eine Programmieraufgabe gilt als bestanden, wenn alle bewerteten Teilaufgaben bestanden sind.
7. Sofern die Aufgabenstellenstellung nichts gegenteiliges besagt, dürfen keine weiteren `include` Direktiven verwendet werden, d.h., es dürfen keine zusätzlichen Bibliotheksfunktionen verwendet werden. Eigene Funktionen zu implementieren und verwenden ist hingegen legitim und häufig eine gute Idee für besser lesbaren Code.

Hinweise zum Repository für das Semester

Die Abgaben werden weiterhin mit Git durchgeführt. Nutzen Sie dazu das selbe Repository wie für die Abgaben des Programmierkurses. Eine Anleitung finden Sie in unserem [ISIS-Kurs](#) auf dem Aufgabenblatt 10 des Programmierkurses.

Aufgabe 1 Implementierung Insertion Sort (bewertet)

Implementieren Sie anhand des Pseudocodes in Listing 1 und der Vorlesungsfolien die Funktion `insertion_sort()` in C. Beachten Sie dabei, dass gemäß Konvention Arrayindizes in Pseudocode bei 1 beginnen, in C jedoch bei 0. Passen Sie die Indizes in Ihrer Implementierung entsprechend an.

Listing 1: Pseudocode Insertion Sort

```
1 InsertionSort(Array A):
2   for j ← 2 to length(A) do
3     key ← A[j]
4     i ← j - 1
5     while i > 0 and A[i] > key do
6       A[i+1] ← A[i]
7       i ← i - 1
8     A[i+1] ← key
```

Die Funktion bekommt als Eingabeparameter ein Integer-Array, das sortiert zurückgegeben werden muss. Die zu sortierenden Zahlen werden aus einer Datei eingelesen. Verwenden Sie dazu die in der Datei `arrayio.c` vorgegebene Funktion `read_array_from_file`. Geben Sie am Ende des Programms das sortierte Array mit der Funktion `print_array` aus.

Machen Sie sich zunächst mit der Signatur der vorgegebenen Funktionen vertraut, um diese korrekt aufzurufen:

- `int read_array_from_file(int array[], size_t size, char* filename)`
Diese Funktion liest eine Folge von maximal `size` vielen Zahlen aus der Datei mit dem Namen `filename` ein und fügt sie in das Array `array` ein.
- `void print_array(int array[], int len)`
Diese Funktion gibt die ersten `len` vielen Einträge des Arrays `array` mittels `printf` aus.

In unserem [ISIS-Kurs](#) ist ein Beispiel mit zu sortierenden Zahlen in der Datei `zahlen_insertionsort.txt` hinterlegt. Die Werte in dieser Datei dienen nur als Illustrationen. Teste Dein Programm also mit unterschiedlichen Eingaben. Halte Dich dabei an das Format der Datei `zahlen_insertionsort.txt`.

Das folgende Listing zeigt Ihnen einen beispielhaften Programmaufruf:

Listing 2: Programmbeispiel

```
1 > clang -std=c11 -Wall introprog_insertionsort.c arrayio.c \  
2     -o introprog_insertionsort  
3 > ./introprog_insertionsort zahlen_insertionsort.txt  
4 Unsortiertes Array: 72 -100 1 10 23 -1 55 9 48 2 -53 5  
5 Sortiertes Array: -100 -53 -1 1 2 4 9 10 23 48 55 72
```

Nutze zur Lösung der Aufgabe die Vorgaben aus unserem [ISIS-Kurs](#). Füge Deine Lösung als Datei `introprog_insertionsort.c` im entsprechenden Abgabebereich in Dein persönliches Repository ein und übertrage die Lösung an die Abgabepattform.