



Onlineprüfung Datenverarbeitung SoSe 2020

Onlineprüfung für DV (SoSe 2020) - Datenverarbeitung im Studiengang MB (Maschinenbau - Bachelor).

Hilfsmittel: DV-Unterlagen & Lehrbücher!

Viel Erfolg!

Hinweise zur Abgabe

Dateiname und -format: **NACHNAME_VORNAME_MATRIKELNUMMER_AUFGABE_X.cpp**

Dateiinhalte: **Quellcode**

Zeit für die Bearbeitung: **90 Minuten**

Zeit für die Abgabe: **5 Minuten**

Gesamtzeit: 95 Minuten

Verspätete Abgaben werden mit 0 Punkten bewertet!

Falsche Dateiformate oder Inhalte werden mit 0 Punkten bewertet!

Verwenden Sie nur in der Vorlesung/Übung erlernte Inhalte.

Falls Sie Schwierigkeiten bei der Abgabe haben, übersenden Sie Ihren Quellcode oder ein leserliches Foto Ihres Quellcodes per E-Mail an mich! (o.uenlue@lba.hs-mannheim.de)

Aufgabe 1 (45 Punkte)

Schreiben Sie ein Sortierungsprogramm mit folgender Menüstruktur und Funktionalitäten.

Ist die letzte Ziffer Ihrer Matrikel-Nr. eine gerade Ziffer, so sortieren Sie die Arrays aufsteigend (1, 2, 5, 9, ...). Ist die letzte Ziffer Ihrer Matrikel-Nr. eine ungerade Ziffer, so sortieren Sie die Arrays absteigend (... , 9, 5, 2, 1).

Sortierungssoftware

1. Größe des Arrays festlegen und Zurücksetzen des Arrays (alle Einträge erhalten den Wert -1)
2. Zahlen in das Array über die Tastatur einlesen (alle Zahlen nacheinander)
3. Eine Zahl im Array an einer bestimmten Position speichern (Positionsangabe wird über die Tastatur eingelesen)
4. Zufallszahlen generieren und das gesamte Array automatisch befüllen
5. Sortierung des Arrays mit BubbleSort
6. Sortierung des Arrays mit SelectionSort
7. Suche nach einem Element im Ursprungsarray (Ausgabe: Element 3 steht an Pos. 7)
8. Suche nach einem Element in den sortierten Arrays (Ausgabe: Element 3 steht an Pos. 7)
9. Ausgabe des Ursprungsarrays
10. Ausgabe der sortierten Arrays (BubbleSort und SelectionSort)
11. Ausgabe der Anzahl der notwendigen logischen Vergleiche bis zur vollständigen Sortierung (BubbleSort und SelectionSort)

0. Programm beenden

Ihre Eingabe:

Hinweise zur Bearbeitung:

- Erlaubte Zahlen im Array sind 1-999
- Es darf kein Element doppelt vorkommen
- Die erlaubte Arraylänge ist 2-100
- Verwenden Sie für jeden Menüpunkt (außer „Programm beenden“) mind. eine Funktion
- Das Programm soll nur beendet werden bei Auswahl des Menüpunktes „Programm beenden“
- Verwenden Sie 3 globale Arrays (*arrayUrsprung*, *arrayBubbleSort*, *arraySelectionSort*)
- Die Ausgabe der Arrays soll mit Kommata getrennt werden (z.B. „3, 7, 12, 29, ...“)
- Als Suchfunktion verwenden Sie die lineare Suche

Aufgabe 2 (30 Punkte)

Programmieren Sie eine Hashtabelle. Die Größe der Hashtabelle wird zu Beginn (vor der Anzeige des Menüs) über die Tastatur eingelesen. Mögliche Werte sind dabei 5 – 100.

Verwenden Sie zwei globale Arrays (*arrayElement* als *char*, *arrayHashwert* als *int*).

Zu Beginn soll der Pufferspeicher das Symbol '#' und den Hashwert "-1" enthalten. Ein Speichern in diesem Zustand, soll nicht möglich sein. Dem Benutzer wird in diesem Falle eine Meldung auf der Konsole ausgegeben.

Sobald der Pufferspeicher in die Hashtabelle eingetragen wurde, wird dieser wieder zurückgesetzt auf den Ursprungszustand '#' und "-1".

Im Falle von Kollisionen verwenden Sie das „Lineare Sondieren“.

Bauen Sie in Ihrem Programm folgende Menüstruktur.

Hashing-Software

Eingelesener Buchstabe im Pufferspeicher: 'X', Hashwert: 24

Füllgrad der Hashtabelle: 12/24 --> 50%

1. *Buchstaben in den Pufferspeicher einlesen und Hashwert berechnen*
2. *Eintragung des Pufferspeichers in die Hash-Tabelle*
3. *Löschen einer bestimmten Position in der Hashtabelle*
4. *Ausgabe der Hashtabelle*
0. *Programm beenden*

Das Programm soll nur beendet werden, wenn der Menüpunkt 0 gewählt wird.

Verwenden Sie zur Berechnung des Hashwertes folgende Tabelle.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Beispielausgabe der Hashtabelle

0:

1: A; Hashwert: 1

2:

3: C; Hashwert: 3

...

Aufgabe 3 (15 Punkte)

Sie haben im Rahmen der Veranstaltung die Caesar-Verschlüsselung kennengelernt. Schreiben Sie zwei Funktionen.

- Name der Funktion: *crypt* → Dient dem Verschlüsseln
- Name der Funktion: *decrypt* → Dient dem Entschlüsseln
- Beide Funktionen sollen den Rückgabotyp *char* haben
- Als Parameter erhalten beide Funktionen einen Buchstaben als *char* und einen *int* als key-Value

Schreiben Sie zudem ein Hauptprogramm, dass über die Tastatur einen *char* und einen key-Value einliest und dem Benutzer die Möglichkeit bietet, das *char* zu verschlüsseln oder zu entschlüsseln.