**Name, Vorname:**

**Matrikelnummer:**

Mit der Abgabe in Moodle versichere ich, dass ich die folgenden Ergebnisse selbst erstellt habe.

Geben Sie das Dokument als PDF ab. Behalten Sie die gegebenen Seitenumbrüche bei.  
  
Achten Sie beim Ausfüllen von mehrzeiligen Textfeldern darauf, dass Ihr Text nicht mehr Zeilen aufweist, als das Textfeld hoch ist! Sämtlicher Text, der durch einen Scrollbalken dargestellt werden muss, wird beim Hochladen der PDF-Datei abgeschnitten!  
Sollte dieses Problem auftauchen, versuchen Sie erst die Textgröße zu verkleinern.

# 1. Speichersegmente

1. Tragen Sie ein „X“ in das Kästchen neben den Aussagen ein, die Ihrer Meinung nach wahr sind.  
   Falsche Antworten führen zu Punktabzügen!

|  |  |
| --- | --- |
|  | **Ohne das Heap-Segment und dynamische Speicherallokation sind Programme nicht lauffähig!** |
|  | **Wenn möglich wird es vermieden, da dynamische Allokationen zu Speicherfragmentierungen von ohnehin wenig Arbeitsspeicher führen können** |
|  | **In Ausnahmefällen, in denen ausreichend Arbeitsspeicher zur Verfügung stehen und/oder von einem Betriebssystem verwaltet werden.** |
|  | **Ohne das Heap-Segment ist es nicht möglich, zur Laufzeit auftretende Speicheranforderungen nachzukommen** |

1. **In welchem Speichersegment werden die lokalen Variablen abgelegt?   
   Im -Segment.  
     
     
   Wieviel Bytes benötigen die drei Variablen insgesamt anhand ihres Datentyps?  
   Bytes.   
     
   Wieviel Bytes reserviert der Compiler für die drei Variablen?  
   Bytes.**
2. Tragen Sie ein „X“ in das Kästchen neben den Aussagen ein, die Ihrer Meinung nach wahr sind.  
   Falsche Antworten führen zu Punktabzügen!

|  |  |
| --- | --- |
|  | **Die Rücksprungadresse der Funktion könnte aus Speichermangel nicht mehr gespeichert werden, welches den Programmfluss durcheinander bringt** |
|  | **Es treten keine Probleme auf, da Rekursionen besonders speicherschonend sind** |
|  | **Das Stack-Segment könnte in das Heap/BSS-Segment überlaufen, wenn viele lokale Variablen genutzt werden** |
|  | **Es passiert nichts, wenn der Stack groß genug dimensioniert ist** |

1. **In welchem Speichersegment werden die initialisierten, globalen Variablen abgelegt?   
   Im -Segment.  
     
     
   In welchem Speichersegment werden die nicht-initialisierten, globalen Variablen abgelegt?   
   Im -Segment.**

**Sie haben folgenden Codeschnipsel gegeben:**short int allowedPortNumbers[32] = {20, 21, 22, 80, 8080};  
 **In welchem Speichersegment erwarten Sie diese Variable?  
Im –Segment, da die Variable initialisierte und nicht-initialisierte Daten enthält.**

1. **Im -Segment**

# 2. Optimierungsstufen

1. **Optimierungsstufe =**
2. **Optimierungsstufe =**
3. **Welche Optimierungsstufe führte zur kürzesten Ausführzeit (real)?  
   Optimierungsstufe   
   .  
     
     
   Erläutern Sie Ihre Annahme, warum die von Ihnen ermittelte Optimierungsstufe zur kürzesten Laufzeit führte:**
4. **Wie wirkt sich die Optimierungsstufe auf Ihre lokalen Variablen aus?   
   Warum wurde die Variable C komplett aus dem Code entfernt?**

**Warum wurde der Zugriff der globalen Variable nicht von der Optimierung verändert? Erläutern Sie kurz Ihre Annahme:**

1. **Wie wurde das Laufzeitverhalten des Programms mit der Optimierungsstufe 2 im Vergleich zur Optimierungsstufe 1 verbessert? Erläutern Sie hierzu, wie der Funktionsaufruf aus der main()-Funktion verändert wurde:  
     
     
   Was ändert sich an der main()-Funktion, wenn man die Optimierungsstufe 's' verwendet (im Vergleich zu Optimierungsstufe 2)?   
     
     
   Wenn Sie die Laufzeit des Programms unter den Optimierungsstufen 2 und 's' vergleichen, würde welche Optimierungsstufe den potentiell schnelleren Code erzeugen und warum?**