

Klausur zu Maschinennahe Programmierung

10.7.2002

Nachname:
Vorname:
Matrikelnummer:

Punkteverteilung

Aufgabe	Punkte	erreicht
1	8	
2	6	
3	7	
4	4	
5	8	
6	7	
Summe	40	
Note	–	

Bitte auf die Rückseiten schreiben oder Zusatzblätter abgeben, wenn der Platz nicht reicht!

Aufg.1) Geben Sie den Inhalt der Register AX, BX, CX, DX nach der Ausführung der untenstehenden Befehle in hexadezimaler Form an. Die Tabelle ist nur für Ihre Notizen gedacht und muss nicht ausgefüllt werden.

Befehl	AX	BX	CX	DX
mov ax,3				
ror al,3				
mov bx,1				
rol bx,2				
add bx,ax				
or bx,2				
mov cx,3				
imul cx,cx				
xor dx,dx				
div cx				
Ergebnis				

Aufg.2) Schreiben Sie in Assembler eine Prozedur, die zwei Bitmuster vergleicht und feststellt, welches die niedrigste Bitposition ist, auf der die beiden Bitmuster übereinstimmen.

Beispiel: Bei den Bitmustern 00110011 und 01011100 ist die niedrigste übereinstimmende Bitposition die Bitposition 4.

Die Schnittstelle soll sein:

Vor Aufruf:

EAX: Zeiger auf erstes Zeichen des Hauptspeicherabschnittes

EBX: Anzahl der auszugebenden Zeichen

Nach Aufruf:

ECX: Die niedrigste übereinstimmende Bitposition

Aufg.3) Schreiben Sie einen Programmabschnitt, der die Zuweisung $c = \sqrt{a^2 + b^2}$ ausführt. Die Variablen a, b, c sollen 64-Bit-Gleitkommazahlen sein.

Aufg.4) Welche Ausgabe auf den Bildschirm produziert das folgende Programm, wenn es als 16-Bit-DOS-Programm übersetzt und ausgeführt wird? Begründen Sie Ihre Antwort!

```
mov dl,0
mov ah,2
marke: int 21h
inc dl
cmp dl,255
jna marke
```

Aufg.5) Schreiben Sie in Inline-Assembler den Rumpf einer C-Funktion (32-Bit-Umgebung), die eine übergebene Zahl auf einen durch Minimum und Maximum angegebenen Wertebereich limitiert. Die Funktion soll folgendes bewirken:

- Liegt die Zahl zwischen Minimum und Maximum, so bleibt sie unverändert.
- Ist die übergebene Zahl kleiner als das angegebene Minimum, wird sie auf den Wert des Minimums gesetzt.
- Ist die übergebene Zahl größer als das angegebene Maximum, wird sie auf den Wert des Maximums gesetzt.

Der Kopf der Funktion soll sein:

```
void Limitiere(int *zahl, int Maximum, int Minimum) {
```

```
}
```

Aufg.6) Schreiben Sie eine gültige C-Funktion auf, aus der bei der Compilierung mit einem 32-Bit-C-Compiler der nebenstehende Assemblercode erzeugt wird!

```
push ebp
mov ebp,esp
push eax
mov eax,[ebp+8]
sal eax,1
mov eax,[ebp+12]
add eax,ebx
sal eax,2
mov esp,ebp
pop ebp
ret
```