

Nachname: Vorname: MatrNr.:

Klausur: Programmieren II

Hochschule für angewandte Wissenschaften
Fakultät Technik und Informatik, Department Informations- und Elektrotechnik
Prof. Robert Heß, 9.7.2012, Bearbeitungsdauer: 90 Min.
Hilfsmittel: Vorlesungsunterlagen und C/C++ Einführungsbücher (z.B. Kernighan/Ritchie).
Jegliche Art von elektronischen Hilfsmitteln sind untersagt.

Ergebnis: von 100 Punkten Note: Punkte.

Aufgabe 1 (10 Punkte)

Sie planen eine Funktion mit Namen `function` zu erstellen und wollen ihr als Parameter eine Zeichenkette und eine **double**-Variable übergeben. Beide Parameter sollen von der Funktion modifiziert werden und Sie wollen den Rückgabewert der Funktion nicht auswerten. Erstellen Sie den Prototyp (d.h. die Deklaration) dieser Funktion.

.....

Aufgabe 2 (20 Punkte)

Erstellen Sie eine Funktion, die eine Variable vom Typ **double** byteweise hexadezimal auf dem Bildschirm ausgibt. Verwenden Sie dafür eine **union**-Struktur. Hinweis: Der Platzhalter für hexadezimale Zahlen ist `%X`.

```
void showBytes(double value)
{
```

```
}
```

Aufgabe 3 (15 Punkte)

Erstellen Sie ein Programm, welches die gegebene Struktur in eine binäre Datei schreibt:

```
#include <stdio.h>

int main()
{
    struct {
        double x, y, z;
        int n;
    } value = { 1.2, 2.3, 3.4, 4 };


```

```
    return 0;
}
```

Aufgabe 4 (15 Punkte)

Das folgende Programm enthält drei Fehler unterschiedlicher Art. Finden Sie die Fehler, und tragen Sie sie unten in die Liste ein.

```

1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main()
5 {
6     int *data=NULL; // pointer for memory allocation
7     int i;          // local index variable
8
9     // get memory
10    data = malloc(10*sizeof(int));
11    if(data==NULL) {
12        printf("ERROR: not enough memory!\n");
13    }
14
15    // evaluate square numbers
16    for(i=0; i<=10; i++) {
17        data[i] = i*i;
18        printf("%d square is %d\n", i, data[i]);
19    }
20
21    // free memory
22    if(data) free(data);
23
24    // end program
25    return 0;
26 }
```

Nr.	Zeile	Fehler
1.		
2.		
3.		

Aufgabe 5 (20 Punkte)

Betrachten Sie folgendes Programm:

```
#include <stdio.h>
#include <string.h>

void function(char *text , char *pt);

int main()
{
    char *text="ABCDabcd123";    // a text

    // do something special
    printf(">");
    function(text , text);
    printf("<\n");

    // quit program
    return 0;
}

void function(char *text , char *pt)
{
    // what happens here?
    if(*pt) {
        printf("%c" , *pt);
        pt += 2;
        if(pt-text > (int)strlen(text)) pt = text+1;
        function(text , pt);
    }
}
```

1. Was gibt die Funktion auf dem Bildschirm aus?

--

2. Was gibt die Funktion aus, wenn Sie an die Zeichenkette noch eine '4' anhängen, also `char *text="ABCDabcd1234";`?

--

Aufgabe 6 (20 Punkte)

Die folgende Klasse realisiert eine verkettete Liste zum Speichern von Personennamen. Die Methode `number()` ermittelt die Anzahl der gespeicherten Namen und die Methode `decrement()` entfernt das erste Element der Kette. Implementieren Sie diese beiden Funktionen.

```
class cChain {
private:
    struct element {
        char surname[32];
        char forename[32];
        element *next;
    };
    element *start;
public:
    cChain();
    ~cChain();
    void decrement();
    int number();
    // ... and some more ...
};

void cChain::decrement()
{
```

```
}
```

```
int cChain::number()
{
```

```
}
```

Viel Erfolg!