
Aufgaben zur Klausur C im WS 97/98 (IA45)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten

Aufgabe 1:

Gegeben sei das folgende C-Programmstück für die Verarbeitung von binären Bäumen:

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

typedef char * String;

typedef struct node * Bintree;
struct node {
    String info;
    Bintree l;
    Bintree r;
};

#define empty ((Bintree)0)

Bintree mk1(String e) {
    Bintree t;
    t = malloc(sizeof(*t));
    if ( !t )
        exit(1);
    t->info = e;
    t->l = empty;
    t->r = empty;
    return t;
}

int compare(String e1, String e2) {
    int c = strcmp(e1,e2);
    return
        ( c > 0 ) ? 1 : ( c == 0 ) ? 0 : -1;
}

extern Bintree insert(String e1, Bintree t);
```

Implementieren Sie die fehlende *insert* Routine. Diese trägt ein Element, hier eine Zeichenreihe, in einen binären Baum ein, falls das Element noch nicht in dem Baum vorhanden ist.

Nutzen Sie die in dem Programmstück vorgegebenen Datentypen, Makros und Funktionen.

(Lösung auf der nächsten Seite).

Aufgabe 2:

Gegeben sei das folgende Programm:

```
#include <stdio.h>

typedef struct entry *dll;
struct entry {
    int i;
    dll n;
    dll p;
};

struct entry e[3] = {
    {-3, e + 1, &e[2]},
    {0, e + 2, &e[0]},
    {5, e, &e[1] }
};

dll l = e;
dll *pl = &l;
int cnt = 0;

int main (int argc, char *argv[])
{
    printf ("%d %d\n", ++cnt, l->i);
    printf ("%d %d\n", ++cnt, l->p->i);
    printf ("%d %d\n", ++cnt, l->p->p->i);
    printf ("%d %d\n", ++cnt, l->p->p->p->i);
    printf ("%d %d\n", ++cnt, l->n->n->n->i);
    printf ("%d %d\n", ++cnt, l->n->p->p->p->n->n->i);

    printf ("%d %d\n", ++cnt, (l++)->n->i);
    printf ("%d %d\n", ++cnt, (l++)->p->i);
    l = e + 2;
    printf ("%d %d\n", ++cnt, (--l)->p->i);
    printf ("%d %d\n", ++cnt, (--l)->n->i);
    l = e + 1;
    printf ("%d %d\n", ++cnt, l[-1].n->i);
    printf ("%d %d\n", ++cnt, l[0].i);
    printf ("%d %d\n", ++cnt, l[+1].p->i);
    l = e;
    printf ("%d %d\n", ++cnt, (**pl).i);

    return 0;
}
```

