

---

Aufgaben zur Klausur **C** im WS 2003/04 (IA 302)

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 7 Seiten

---

## Aufgabe 1:

Gegeben sei das folgende C-Programm:

```
#include <stdio.h>

#define maxm(x,y) ((x) > (y) ? (x) : (y))

long maxf(long x, long y) { return x > y ? x : y; }

unsigned cnt = 0;

#define eval(x) ( ++cnt, (x) )

int main(void) {
    long f[] = { 0, +2, -4, +6, -8 };
    long res;

    double g[] = { 3.1, 1.5, 4.9 };
    double r;

    cnt = 0;
    res = maxm( maxm( eval(f[1]), eval(f[2]) ),
                eval(f[3]) );

    printf(" res = %ld, cnt = %u\n", res, cnt);

    cnt = 0;
    res = maxf( maxf( eval(f[1]), eval(f[2]) ),
                eval(f[3]) );

    printf(" res = %ld, cnt = %u\n", res, cnt);

    cnt = 0;
    r = maxm( eval(g[1]), eval(g[2]) );

    printf(" r = %3.1f, cnt = %u\n", r, cnt);

    cnt = 0;
    r = maxf( eval(g[1]), eval(g[2]) );

    printf(" r = %3.1f, cnt = %u\n", r, cnt);

    return 0;
}
```

Welche vier Ausgabezeilen erzeugt dieses Programm:

1) .....

2) .....

3) .....

4) .....

---

## Aufgabe 2:

Gegeben sei das folgende C-Programm zur Verarbeitung von Mengen als Bitstrings.

```
#include <stdio.h>

typedef unsigned char Set;
#define SetMax 8

void printSet(Set s) {
    unsigned int i = SetMax;
    while ( i-- != 0 )
        printf("%1u", (unsigned int)((s >> i) & 1));
}

#define PRINT(s) { printSet(s); printf("\n"); }

#define single(i) ( (Set)(1 << (i)) )
#define first(n) (single(n) - 1)
#define interval(n,m) (first(m+1) ^ first(n))

int main(void) {
    Set s1;

    PRINT( (Set)0 );
    PRINT( single(SetMax / 2 - 1) );
    PRINT( (Set)-1 );

    PRINT( interval(3,6) );
    PRINT( interval(0,2) );
    PRINT( interval(0,SetMax) );

    PRINT( 2 | 7 );
    PRINT( 2 || 7 );
    PRINT( first(SetMax / 2 + 1) );

    s1 = interval(1,4) & ~interval(2,6); PRINT(s1);
    s1 = interval(1,4) ^ interval(2,6); PRINT(s1);

    s1 = 2 + 16 + 32;
    s1 ^= s1 & (~s1 + 1); PRINT(s1);

    return 0;
}
```

Die Mengen sind in diesem Beispiel 8 Bits lang, können also die Elemente  $0, 1, \dots, 7$  enthalten. *printSet* gibt eine Menge im Binärformat aus. Die Menge, die nur die 1 enthält würde als 00000010 ausgegeben werden. Das *PRINT* Makro gibt jeweils eine Menge pro Zeile aus.

Welche 12 Ausgabezeilen erzeugt dieses Programm?

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....
- 6) .....
- 7) .....
- 8) .....
- 9) .....
- 10) .....
- 11) .....
- 12) .....

### Aufgabe 3:

Gegeben sei das folgende Programm:

```
#include <stdio.h>

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

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

dll l = e;
dll *pl = &l;

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

    l = e + 2;
    printf ("%d\n", (--l)->p->i);
    printf ("%d\n", (--l)->n->i);

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

    printf ("%d\n", (**pl).i);

    return 0;
}
```

Welche Ausgabezeilen liefert dieses Programm:

- 1) .....
  - 2) .....
  - 3) .....
  - 4) .....
  - 5) .....
  - 6) .....
  - 7) .....
  - 8) .....
  - 9) .....
  - 10) .....
  - 11) .....
  - 12) .....
  - 13) .....
  - 14) .....
-