

Musterlösung der 6. Übung

1. Vollständiger Quelltext:

```
#include <stdio.h>
int main()
{
    int zahl1, zahl2, ergebnis;
    char zeichen;
    int gelesen;

    printf("Bitte die erste Zahl eingeben: ");
    gelesen = scanf("%d", &zahl1);

    if(gelesen != 1)
    {
        printf("Sie haben keine Zahl eingegeben! Programmabbruch.\n");
        return 1;
    }

    printf("Bitte die zweite Zahl eingeben: ");
    gelesen = scanf("%d", &zahl2);

    if(gelesen != 1)
    {
        printf("Sie haben keine Zahl eingegeben! Programmabbruch.\n");
        return 1;
    }

    printf("Bitte geben Sie ein Zeichen ein.\n");
    printf("+ für Addition\n");
    printf("- für Subtraktion\n");
    printf("* für Multiplikation\n");
    printf("/ für Division\n");
    printf("Ihr gewünschter Operator: ");
    /* Problem: Nach dem letzten Einlesen einer Zahl ist im Eingabepuffer noch
       die Eingabetaste, d.h. das scanf("%c") liest immer einen
       Zeilenvorschub.
       Lösungen: z.B. mit scanf() mehrmals Zeichen lesen, bis eins dabei ist,
       was +, -, *, / entspricht, ODER, die Spezifikation von scanf lesen,
       ein Leerzeichen vor %... bedeutet "Leerzeichen, Leerzeilen usw.
       ignorieren". */
    gelesen = scanf(" %c", &zeichen);

    if(gelesen != 1)
    {
        printf("Sie haben kein Zeichen eingegeben! Programmabbruch.\n");
        return 1;
    }
}
```

```

#if 0 /* Alte Version mit if ... else if ... else */
  if(zeichen == '+') ergebnis = zahl1 + zahl2;
  else if(zeichen == '-') ergebnis = zahl1 - zahl2;
  else if(zeichen == '*') ergebnis = zahl1 * zahl2;
  else if(zeichen == '/')
  {
    if(zahl2 == 0)
    {
      printf("Durch 0 darf man nicht teilen, Abbruch.\n");
      return 10;
    }
    ergebnis = zahl1 / zahl2;
  }
else
  {
    printf("Sie haben leider keine Rechenoperation angegeben, Abbruch.\n");
    return 1;
  }
#else
switch(zeichen) /* Achtung: Unbedingt mit break; abschließen, sonst */
{ /* werden auch die nachfolgenden case-Anweisungen mit ausgeführt! */
  case '+': ergebnis = zahl1 + zahl2; break;
  case '-': ergebnis = zahl1 - zahl2; break;
  case '*': ergebnis = zahl1 * zahl2; break;
  case '/': if(zahl2 == 0)
    { printf("Durch 0 darf man nicht teilen, Abbruch.\n");
      return 10; }
    ergebnis = zahl1 / zahl2; break;
  default: printf("Sie haben leider keine Rechenoperation angegeben, "
    "Abbruch.\n");
    return 1;
}
#endif

printf("%d %c %d = %d\n", zahl1, zeichen, zahl2, ergebnis);

return 0;
}

```

2. Vollständiger Quelltext:

```

int fibo_rek(int x)
{
  if(x == 0) return 0;
  if(x == 1) return 1;
  return fibo_rek(x-1) + fibo_rek(x-2);
}

/* Hier ist die Aufgabe bereits gelöst, wir schreiben aber
   noch schnell ein main() zum Testen. */
#include <stdio.h>

```

```

int main()
{
    printf("fibonacci(30) = %d\n", fibo_rek(30) );
    return 0;
}

```

3. Vollständiger Quelltext:

```

int fibo_iter(int x)
{
    int i;
    int ergebnis = 0;
    int vorgaenger = 1;
    int vorvorgaenger = 0;

    if(x == 0) return 0;
    if(x == 1) return 1;

    for(i=2; i<=x; i++)
    {
        ergebnis = vorgaenger + vorvorgaenger;
        vorvorgaenger = vorgaenger;
        vorgaenger = ergebnis;
    }
    return ergebnis;
}

/* Hier ist die Aufgabe bereits gelöst, wir schreiben aber
   noch schnell ein main() zum Testen. */
#include <stdio.h>
int main()
{
    int zahl;
    printf("Bitte eine Zahl eingeben: ");
    scanf("%d", &zahl);

    printf("fibonacci(%d) = %d\n", zahl, fibo_iter(zahl) );
    return 0;
}

```

4. Vollständiger Quelltext:

```

#include <stdio.h>

#define FIBOMAX 35

/* Das ginge alternativ auch: */
/* const int FIBOMAX = 35; */

/* Da die Abfrage an mehreren Stellen durchgeführt werden kann,
   ist es sinnvoll, eine Funktion dafür zu schreiben. */

```

```

void check(int x)
{
    if(x > FIBOMAX)
    {
        printf("Zahl ist größer als die erlaubte %d, Abbruch!\n", FIBOMAX);
        exit(-1);
    }
    else if(x<0)
    {
        printf("Von negativen Zahlen kann keine Fibonacci-Funktion "
               "berechnet werden!\n");
        exit(-1);
    }
}

int fibo_rek(int x)
{
    check(x);
    if(x == 0) return 0;
    if(x == 1) return 1;
    return fibo_rek(x-1) + fibo_rek(x-2);
}

int fibo_iter(int x)
{
    int i;
    int ergebnis = 0;
    int vorgaenger = 1;
    int vorvorgaenger = 0;

    check(x);
    if(x == 0) return 0;
    if(x == 1) return 1;

    for(i=2; i<=x; i++)
    {
        ergebnis = vorgaenger + vorvorgaenger;
        vorvorgaenger = vorgaenger;
        vorgaenger = ergebnis;
    }
    return ergebnis;
}

int main()
{
    int zahl;
    printf("Bitte eine Zahl eingeben: ");
    scanf("%d", &zahl);
    printf("fibo_rek(%d) = %d\n", zahl, fibo_rek(zahl) );
    printf("fibo_iter(%d) = %d\n", zahl, fibo_iter(zahl) );
    return 0;
}

```