```
/* ****************************************************************************** */
/* Musterl“osung ”Ubung 2, Aufgabe 1 */
/* ****************************************************************************** */

#include<stdio.h>

/* ——————————————————————————————————- */
/* Funktionsdeklaration                                             */
/* ——————————————————————————————————- */
/* die Funktion arith1: sie liefert das arithmetische Mittel zur“uck */
float arith1(int a, int b)                                              arith1
{
  return ( ((float)(a+b)) / 2);

}
/* ANMERKUNG:
   Die Variablen a und b dieser Funktion sind lokale Variablen und somit
   nicht diesselben, wie die des Hauptprogramms, d.h. wenn man in arith1
   den Wert von a ver“andert, so hat dies keine Auswirkungen auf den Wert
   des a’s aus dem Hauptprogramm.
   Testen sie dies, indem sie in arith1 das a veraendern (zB: a=42;) und
   im Hauptprogramm das a vor und nach dem Aufruf von arith1 ausgeben.
   Ausserdem k”onnte man noch das return anderes schreiben
   mit implizitem Cast:
        return ( (a+b)/2.0) ;
*/

/* die Funktion airth2: sie gibt das arithmetische Mittel aus */
void arith2(int a, int b)                                               arith2
{
  printf("arith2: Das arithmetische  Mittel von %d und %d ist %f.\n",
    a, b, (float)(a+b)/2);
}


/* ——————————————————————————————————- */
/* Das Hauptprogramm                                                */
/* ——————————————————————————————————- */
int main (void)                                                         main
{
  /* Variablendeklaration */
  int a;
  int b;
  float mittel;

  /* Einlesen der Variablen */
  printf("Bitte gebe eine ganze Zahl ein: ");
  scanf("%d", &a);
  printf("Bitte gebe eine ganze Zahl ein: ");
  scanf("%d", &b);
```



```

  /* Aufruf von arith1() */
  mittel = arith1(a,b);
  printf("arith1: Das arithmetische  Mittel von %d und %d ist %f.\n",
    a, b, mittel);

  /* Aufruf von arith2() */
  arith2(a,b);

  return 0;  /*** R“uckgabewert ans Betriebssystem ***/
}


/* ****************************************************************************** */
/* Musterl“osung ”Ubung 2, Aufgabe 2 */
/* ****************************************************************************** */

#include<stdio.h>


/* ——————————————————————————————————- */
/* gloable Varibalen                                                */
/* ——————————————————————————————————- */
float a;
float b;


/* ——————————————————————————————————- */
/* Funktionsdeklaration                                             */
/* ——————————————————————————————————- */
/* die Funktion min1: sie bekommt zwei float Werte als Argumente */
float min1(float a, float b)                                            min1
{
  /* Versuch: “Andern sie hier die Werte von a und b und pr”ufen sie,
     was mit den globalen Variablen a und b passiert.
  */
  if (a < b) return a;
  return b;
}

/* die Funktion min2: sie hat keine Argumente */
float min2()                                                            min2
{
  if (a < b) return a;
  return b;
}


/* ——————————————————————————————————- */
/* Das Hauptprogramm                                                */
```

```
/* ——————————————————————————————- */
int main (void)                                                    main
{
  /* Variablendeklaration */
  float minimum;

  /* Einlesen der Variablen */
  printf("Bitte gebe eine Fliesskommazahl ein: ");
  scanf("%f", &a);
  printf("Bitte gebe noch eine Fliesskommazahl ein: ");
  scanf("%f", &b);

  /* Aufruf von min1() */
  minimum = min1(a,b);
  printf("min1: Das Minimum von %f und %f ist %f.\n",
    a, b, minimum);

  /* Aufruf von min2() */
  minimum = min2();
  printf("min2: Das Minimum von %f und %f ist %f.\n",
    a, b, minimum);

  return 0;  /*** R“uckgabewert ans Betriebssystem ***/
}


/* ************************************************************************** */
/* Musterl“osung ”Ubung 2, Aufgabe 2 */
/* ************************************************************************** */

#include<stdio.h>


/* ——————————————————————————————- */
/* Funktionsdeklaration                                    */
/* ——————————————————————————————- */
/* iterative Berechnung der Fakult“at */
int iter_faq(int n)                                                iter_faq
{
  int faq=1;

  while (n>=1)
  {
    faq = faq * n;
    n−−;
  }
  /* oder superkurz und superun“ubersichtlich: while (n) faq *= n–; */

  return faq;
}
```



```
/* rekursive Berechnung der Fakult“at */
int rek_faq(int n)                                                 rek_faq
{
  if (n==1) return 1;                 /* Rekursionsende bei 1 */
  if (n > 1) return n*rek_faq(n−1);  /* Rekursionsschritt wenn n groesser als 1 */
  return 1;                           /* Falls n<=0 ist gib eine 1 zur“uck */
}


/* ——————————————————————————————- */
/* Das Hauptprogramm                                           */
/* ——————————————————————————————- */
int main (void)                                                    main
{
  /* Variablendeklaration */
  int n;
  int fakultaet;

  /* Einlesen der Variablen */
  printf("Bitte gebe eine natuerliche Zahl ein: ");
  scanf("%d", &n);

  /* Aufruf von iter_faq() */
  fakultaet = iter_faq(n);
  printf("iter_faq: Die Fakultaet von %d ist %d.\n", n, fakultaet);

  /* Aufruf von rek_faq() */
  fakultaet = rek_faq(n);
  printf("rek_faq:  Die Fakultaet von %d ist %d.\n", n, fakultaet);

  return 0;  /*** R“uckgabewert ans Betriebssystem ***/
}
```