```c
#include <stdio.h>
#include <math.h>

#include <Eigene Pakete/ausgabe.h>
```

```c
#include "includetest1.h"

int Anzahl_der_Daten = 0;
char Text[128];

const char *Ausgabe_Textdatei = "Ausgabe Aufgabe 1 test1.txt";

typedef struct point POINT;
struct point
{
  double x;
  double y;
};

struct triangle
{
  struct point pt0;
  struct point pt1;
  struct point pt2;
};

double dist(struct point p1, struct point p2)
{
  p1.x -= p2.x;
  p1.y -= p2.y;
  printf("p1 = (%f,%f)\n", p1.x, p1.y);
  sprintf(Text,"p1 = (%f,%f)\r\n", p1.x, p1.y);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);

  return(sqrt(p1.x*p1.x + p1.y*p1.y));
}


struct point set_point(double x, double y)
{
  struct point temp;
  temp.x = x;
  temp.y = y;
  return temp;
}

void print_point(char *name, struct point p)
{
  printf("%s = (%f,%f)\n", name, p.x, p.y);
  sprintf(Text,"%s = (%f,%f)\r\n", name, p.x, p.y);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
}

void point_example(void)
{
  double h;
  struct point  p1 = {0.0, 2.0}, p2;
  p2 = set_point(1.0, 1.0);

  print_point("p1", p1);
  print_point("p2", p2);
  printf("dist(p1,p2) = %f\n", h = dist(p1, p2));
  sprintf(Text,"dist(p1,p2) = %f\r\n", h);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
  print_point("p1", p1);
  print_point("p2", p2);
}

struct point barycenter(struct triangle t)
{
  t.pt0.x += t.pt1.x + t.pt2.x;
```

```
67    t.pt0.x *= 1.0/3.0;
68    t.pt0.y += t.pt1.y + t.pt2.y;
69    t.pt0.y *= 1.0/3.0;
70    return(t.pt0);
71  }
72
73  void print_triangle(char *name, struct triangle t)
74  {
75    char  namept[128];
76
77    sprintf(namept, "%s.p0", name);
78    print_point(namept, t.pt0);
79    sprintf(namept, "%s.p1", name);
80    print_point(namept, t.pt1);
81    sprintf(namept, "%s.p2", name);
82    print_point(namept, t.pt2);
83  }
84
85  void triangle_example()
86  {
87    struct triangle t = {{0.0,0.0}, {1.0,0.0}, {0.0,1.0}};
88
89    print_triangle("triangle", t);
90    print_point("barycenter", barycenter(t));
91    print_triangle("triangle", t);
92  }
93
94  int main()
95  {
96    point_example();
97    triangle_example();
98
99    return(0);
100 }
```

```
p1 = (0.000000,2.000000)
p2 = (1.000000,1.000000)
p1 = (-1.000000,1.000000)
dist(p1,p2) = 1.414214
p1 = (0.000000,2.000000)
p2 = (1.000000,1.000000)
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
barycenter = (0.333333,0.333333)
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
```

```c
#include <stdio.h>
#include <math.h>

#include <Eigene Pakete/ausgabe.h
```

```c
#include "includetest2.h"

int Anzahl_der_Daten = 0;
char Text[128];

const char *Ausgabe_Textdatei = "Ausgabe Aufgabe 1 test2.txt";

struct point
{
  double x;
  double y;
};

struct triangle
{
  struct point pt0;
  struct point pt1;
  struct point pt2;
};


struct point set_point(double x, double y)
{
  struct point temp;
  temp.x = x;
  temp.y = y;
  return temp;
}

void print_point(char *name, struct point *p)
{
  printf("%s = (%f,%f)\n", name, p->x, p->y);
/* <=>  printf("%s = (%f,%f)\n", name, (*p).x, (*p).y);  */
  sprintf(Text,"%s = (%f,%f)\r\n", name, p->x, p->y);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
}

/*  !!!wrong implementation!!!                                  */

double dist_wrong(struct point *p1, struct point *p2)
{
  (*p1).x -= (*p2).x;  /*  <=> p1->x -= p2->x */
  p1->y -= p2->y;
  print_point("p1 in dist", p1);
  return(sqrt(p1->x*p1->x + p1->y*p1->y));
}

double dist_right(struct point *p1, struct point *p2)
{
  struct point p;
  p.x = p1->x - p2->x;
  p.y= p1->y - p2->y;

  return(sqrt(p.x*p.x + p.y*p.y));
}

void print_point_better(const char *name, const struct point *p)
{
  printf("%s = (%f,%f)\n", name, p->x, p->y);
  sprintf(Text,"%s = (%f,%f)\r\n", name, p->x, p->y);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
}

double dist_better(const struct point *p1, const struct point *p2)
{
  struct point p;
```

```
67    p.x = p1->x - p2->x;
68    p.y= p1->y - p2->y;
69
70    return(sqrt(p.x*p.x + p.y*p.y));
71  }
72
73  void point_example(void)
74  {
75    double h;
76    struct point  p1 = {0.0, 2.0}, p2;
77    p2 = set_point(1.0, 1.0);
78
79    print_point("p1", &p1);
80    print_point("p2", &p2);
81    printf("dist(p1,p2) = %f\n", h = dist_wrong(&p1, &p2));
82    sprintf(Text,"dist(p1,p2) = %f\r\n", h);
83    Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
84    print_point("p1", &p1);
85    print_point("p2", &p2);
86    p1 = set_point(0.0,2.0);
87    printf("dist(p1,p2) = %f\n", dist_right(&p1, &p2));
88    sprintf(Text,"dist(p1,p2) = %f\r\n", dist_right(&p1, &p2));
89    Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
90    print_point("p1", &p1);
91    print_point("p2", &p2);
92  }
93
94  struct point barycenter(const struct triangle *t)
95  {
96    struct point p;
97    p.x = 1.0/3.0*(t->pt0.x + t->pt1.x + t->pt2.x);
98    p.y = 1.0/3.0*(t->pt0.y + t->pt1.y + t->pt2.y);
99    return(p);
100 }
101
102 void print_triangle(const char *name, const struct triangle *t)
103 {
104   char  namept[128];
105
106   sprintf(namept, "%s.p0", name);
107   print_point_better(namept, &t->pt0);
108   sprintf(namept, "%s.p1", name);
109   print_point_better(namept, &t->pt1);
110   sprintf(namept, "%s.p2", name);
111   print_point_better(namept, &t->pt2);
112 }
113
114 void triangle_example()
115 {
116   struct triangle t = {{0.0,0.0}, {1.0,0.0}, {0.0,1.0}};
117   struct triangle *pt;
118   struct point    b, *pb = &b;
119
120   pt = &t;
121   print_triangle("triangle", pt);
122   b = barycenter(pt);
123   print_point("barycenter", pb);
124   print_triangle("triangle", pt);
125 }
126
127 int main()
128 {
129   point_example();
130   triangle_example();
131
132   return(0);
```

```
133  }
```

```
p1 = (0.000000,2.000000)
p2 = (1.000000,1.000000)
p1 in dist = (-1.000000,1.000000)
dist(p1,p2) = 1.414214
p1 = (-1.000000,1.000000)
p2 = (1.000000,1.000000)
dist(p1,p2) = 1.414214
p1 = (0.000000,2.000000)
p2 = (1.000000,1.000000)
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
barycenter = (0.333333,0.333333)
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
```

```c
#include <stdio.h>
#include <math.h>

#include <Eigene Pakete/ausgabe.h>
```

```c
#include "includetest3.h"

int Anzahl_der_Daten = 0;
char Text[128];

const char *Ausgabe_Textdatei = "Ausgabe Aufgabe 1 test3.txt";

struct point
{
  double x;
  double y;
};

struct triangle
{
  struct point pt[3];
};


struct point set_point(double x, double y)
{
  struct point temp;
  temp.x = x;
  temp.y = y;
  return temp;
}

void print_point(const char *name, const struct point *p)
{
  printf("%s = (%f,%f)\n", name, p->x, p->y);
  sprintf(Text,"%s = (%f,%f)\r\n", name, p->x, p->y);
  Anzahl_der_Daten = speichern_in_Datei(Anzahl_der_Daten,Ausgabe_Textdatei,Text);
}

double dist(const struct point *p1, const struct point *p2)
{
  struct point p;
  p.x = p1->x - p2->x;
  p.y= p1->y - p2->y;

  return(sqrt(p.x*p.x + p.y*p.y));
}

struct point barycenter(const struct triangle *t)
{
  struct point p = {0.0, 0.0};
  int          i;

  for (i = 0; i < 3; i++)
  {
    p.x += t->pt[i].x;
    p.y += t->pt[i].y;
  }

  p.x /= 3.0;
  p.y /= 3.0;

  return(p);
}

void print_triangle(const char *name, const struct triangle *t)
{
  char  namept[128];
  int   i;

  for (i = 0; i < 3; i++)
```

```
67    {
68      sprintf(namept, "%s.p%d", name, i);
69      print_point(namept, t->pt+i);
70  /* <=>   print_point(namept, &t->pt[i]); */
71    }
72  }
73
74  void triangle_example()
75  {
76    struct triangle t = {{{0.0,0.0}, {1.0,0.0}, {0.0,1.0}}};
77    struct triangle *pt;
78    struct point    b, *pb = &b;
79
80    pt = &t;
81    print_triangle("triangle", pt);
82    b = barycenter(pt);
83    print_point("barycenter", pb);
84    print_triangle("triangle", pt);
85  }
86
87  int main()
88  {
89    triangle_example();
90
91    return(0);
92  }
```

```
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
barycenter = (0.333333,0.333333)
triangle.p0 = (0.000000,0.000000)
triangle.p1 = (1.000000,0.000000)
triangle.p2 = (0.000000,1.000000)
```