```c
/*****************************************/
/* Introduction to Compiler Construction */
/*                                       */
/* Christoph Kirsch                      */
/* University of Salzburg                */
/*                                       */
/* Example Records                       */
/*****************************************/

#include <stdlib.h>
#include <stdio.h>

main() {
  // optional
  struct record_t {
    int f;
    int g;
  };
  struct record_t r1;
  r1.f = 1;
  r1.g = 2;

  // required
  struct record_t *r2;
  r2 = malloc(sizeof(struct record_t));
  r2->f = 3;
  r2->g = 4;

  // required
  struct record_of_record_t;
  struct record_of_record_t {
    int f;
    struct record_of_record_t *g;
  };
  struct record_of_record_t *r3;
  r3 = malloc(sizeof(struct record_of_record_t));
  r3->f = 5;
  r3->g = r3;

  // required
  typedef int *array_t;
  array_t a1;
  a1 = malloc(10 * sizeof(int));
  struct record_of_array_t {
    int f;
    array_t g;
  };
  struct record_of_array_t *r4;
  r4 = malloc(sizeof(struct record_of_array_t));
  r4->f = 6;
  r4->g = a1;

  // required
  typedef struct record_t * *array_of_record_references_t;
  array_of_record_references_t a2;
  a2 = malloc(2 * sizeof(struct record_t *));
  a2[1] = r2;

  printf("a2[1]->g: %d\n", a2[1]->g);
  // optional
  typedef struct record_t *array_of_records_t;
  array_of_records_t a3;
  a3 = malloc(2 * sizeof(struct record_t));
  a3[1] = r1;

  printf("a3[1].g: %d\n", a3[1].g);
}
```