```
/*****************************************/
/* Introduction to Compiler Construction */
/*                                       */
/* Christoph Kirsch                      */
/* University of Salzburg                */
/*                                       */
/* access to record fields               */
/*****************************************/

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

struct record_t {
  int f;
  int g;
};

typedef struct record_t * *array_of_record_references_t;

struct record_of_array_t {
  int u;
  array_of_record_references_t v;
  int w;
};

typedef struct record_of_array_t * *array_of_record_of_array_references_t;

int i;
int j;
int k;

array_of_record_of_array_references_t s;

main() {
  i = 0;
  // ADDI 1, 0, 0 or MOVI 1, 0, 0
  // STW 1, 28, -4

  j = 0;
  // ADDI 1, 0, 0 or MOVI 1, 0, 0
  // STW 1, 28, -8

  k = 0;
  // ADDI 1, 0, 0 or MOVI 1, 0, 0
  // STW 1, 28, -12

  s = malloc(2 * sizeof(struct record_of_array_t *));
  s[0] = malloc(sizeof(struct record_of_array_t));
  s[0]->v = malloc(4 * sizeof(struct record_t));
  s[0]->v[0] = malloc(sizeof(struct record_t));
  s[0]->v[1] = malloc(sizeof(struct record_t));
  s[0]->v[2] = malloc(sizeof(struct record_t));
  s[0]->v[3] = malloc(sizeof(struct record_t));
  s[1] = malloc(sizeof(struct record_of_array_t));
  s[1]->v = malloc(4 * sizeof(struct record_t));
  s[1]->v[0] = malloc(sizeof(struct record_t));
  s[1]->v[1] = malloc(sizeof(struct record_t));
  s[1]->v[2] = malloc(sizeof(struct record_t));
  s[1]->v[3] = malloc(sizeof(struct record_t));
```

```
    k = s[i]->u;
    // LDW 1, 28, -4
    // MULI 1, 1, 4: unlike MULI 1, 1, 40
    // LDW 2, 28, -16: deref from VAR_MODE into REG_MODE
    // ADD 2, 2, 1: index from REG_MODE into REF_MODE
    // LDW 2, 2, 0: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 2, 2, 0: load from REF_MODE into REG_MODE: unlike LDW 2, 1, -92
    // STW 2, 28, -12

    k = s[1]->w;
    // LDW 1, 28, -16: deref from VAR_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 1*4: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 8: load from REF_MODE into REG_MODE: unlike LDW 1, 0, -16
    // STW 1, 28, -12

    k = s[i]->v[j]->f;
    // LDW 1, 28, -4
    // MULI 1, 1, 4: unlike MULI 1, 1, 40
    // LDW 2, 28, -16: deref from VAR_MODE into REG_MODE
    // ADD 2, 2, 1: index from REG_MODE into REF_MODE
    // LDW 2, 2, 0: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 28, -8
    // MULI 1, 1, 4: unlike MULI 2, 2, 8
    // LDW 2, 2, 4: deref from REF_MODE into REG_MODE
    // ADD 2, 2, 1: index from REG_MODE into REF_MODE
    // LDW 2, 2, 0: load from REF_MODE into REG_MODE: unlike LDW 1, 2, -88
    // STW 2, 28, -12

    k = s[1]->v[2]->g;
    // LDW 1, 28, -16: deref from VAR_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 1*4: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 4: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 2*4: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 4: load from REF_MODE into REG_MODE: unlike LDW 1, 0, -28
    // STW 1, 28, -12

    s[0]->v[i]->g = k;
    // LDW 1, 28, -16: deref from VAR_MODE into REF_MODE (via REG_MODE)
    // LDW 1, 1, 0*4: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 2, 28, -4
    // MULI 2, 2, 4: unlike MULI 1, 1, 8
    // LDW 1, 1, 4: deref from REF_MODE into REG_MODE
    // ADD 1, 1, 2: index from REG_MODE into REF_MODE
    // LDW 1, 1, 0: deref from REF_MODE into REF_MODE (via REG_MODE)
    // LDW 2, 28, -12
    // STW 2, 1, 4: unlike STW 2, 1, -84
}
```