

## Oefenzitting 6 - Functies III (Records)

### 1.12 Records in DRAMA

#### Oefening 1

Vertaal het volgende C programma.

```
struct rationaalgetal {
    int teller;
    int noemer;
    int gehdeel;
};
// ===== Eerder vertaald: =====
struct rationaalgetal maakrationaal(register int gehdeel, int teller,
                                   int noemer) { ... } // zie huistaak
void drukrationaal (struct rationaalgetal g) { ... } // zie huistaak vorige week
int kgv(int a, int b) { ... } // kleinst gemeen veelvoud, zie oefenzitting 3
int ggd(int a, register int b) { ... } // grootst gemene deler, zie oefenzitting 3
// =====

struct rationaalgetal verschilrationaal (struct rationaalgetal a,
    register struct rationaalgetal * b) {
    struct rationaalgetal res;
    register int k, fac;
    // trek gehele delen van elkaar af
    res.gehdeel = a.gehdeel - b->gehdeel;
    // noemer is kleinste gemeen veelvoud van de 2 noemers
    k = kgv(a.noemer, b->noemer);
    res.noemer = k;
    // teller is het verschil van de tellers (vermenigvuldigd met een factor)
    fac = k / a.noemer;
    res.teller = fac * a.teller;
    fac = k / b->noemer;
    res.teller -= fac * b->teller;
    // vereenvoudig teller / noemer
    fac = ggd (res.teller, res.noemer);
    if (fac != 1) {
        res.teller /= fac;
        res.noemer /= fac;
    }
    return (res);
}
```

```

struct rationaalgetal x; // globale variabele
main () {
    struct rationaalgetal y, v;
    x = maakrationaal(20, 7, 10);
    y = maakrationaal(5, 1, 5);
    v = verschilrationaal (x, &y);
    drukrationaal(v); // 20 7/10 - 5 1/5 = 15 1/2
}

```

## Oefening 2

Vertaal het volgende C-programma.

```

struct rec {
    int len;
    int rij[5];
};
struct rec mkRec (register int * r, int n) {
    struct rec tmp;
    register int i;
    tmp.len = n;
    for (i = 0; i < n; i++) {
        tmp.rij[i] = r[i];
    }
    return (tmp);
}
void rec2Rij (struct rec s, int r[]) {
    register int i, * ptr;
    ptr = s.rij;
    for (i = 0; i < s.len; i++) {
        r[i] = *ptr++;
    }
}
int grij[] = {0, 0, 0, 0, 0};
main () {
    int rij[] = {1, 2, 3, 4, 5};
    struct rec REC;
    register int i;
    REC = mkRec(rij, 3);
    rec2Rij (REC, grij);
    for (i = 0; i < 5; i++) {
        printint(grij[i]);
    }
}

```

### Oefening 3

Bestudeer het volgende C-programma. Waarom loopt het fout?

```
struct loc {
    int x, y;
};

struct loc * mkLoc (int a, b) {
    struct loc l;
    l.x = a;
    l.y = b;
    return (&l);
}

void druk (struct loc l) {
    printint (l.x, l.y);
}

main () {
    struct loc mijnLoc, * wzLoc;
    mijnLoc.x = 10;
    mijnLoc.y = 20;
    wzLoc = mkLoc (2, 3);
    druk (mijnLoc);
    mijnLoc = *wzLoc;
    druk (mijnLoc); // Zou 2 3 moeten afdrukken!
}
```