

## Huistaak - Records

Nota:

1. indien 'register' in de declaratie van de lokale variabele of formele parameter staat, dan moet die variabele of parameter in een register worden bijgehouden (resp. doorgegeven); anders moet deze variabele op de stapel worden bijgehouden (resp. via de stapel worden doorgegeven);
2. een functie geeft zijn resultaat terug via R0.

### Oefening 1

Wat is de vertaling van de volgende C-code?

```
struct deel {
    int hoofdstuk;
    int sectie;
    int paragraaf;
    int gestudeerd;
};
struct deel socs[100];

main()
{
    register int k, * wz;
    int j;

    ...
    socs[k].paragraaf = 2;
    socs[8].sectie = socs[j-2].sectie;

    // initialiseer elk deel als (nog) niet gestudeerd
    for (wz = socs; wz < &socs[100]; wz++)
        wz->gestudeerd = 0;
    ...
}
```

## Oefening 2

Wat doet dit C-programma? Wat is de vertaling van het volgende C-programma?

```
struct rationaalgetal
{
    int teller;
    int noemer;
    int gehdeel;
};

void leesrationaal(register struct rationaalgetal * g)
{
    g->gehdeel = getint();
    g->teller = getint();
    g->noemer = getint();
}

void drukrationaal (struct rationaalgetal g)
{
    printint(g.gehdeel, g.teller, g.noemer);
}

main ()
{
    struct rationaalgetal q;

    leesrationaalgetal(&q);
    while (q.noemer != 0)
    {
        drukrationaal(q);
        leesrationaal(&q);
    }
}
```

### Oefening 3

Hoeveel geheugenregisters moet men reserveren voor de variabele robot?  
Merk op: een veld in een record mag om het even welk type hebben of zelfs een rij zijn.

```
struct plaats {
    int x;
    int y;
};
struct positie {
    struct plaats loc;
    int hgt;
};
struct tool {
    int nr;
    int vrij;
    int gewicht;
};
struct object {
    struct positie pos;
    int eigenschap[10];
    struct positie route[20]; // d.i. rij van 20 posities
    struct tool (* tools)[10]; // d.i. wijzer naar rij van 10 tool-structs
    struct tool * mijntools[5]; // d.i. rij van 5 wijzers naar tool-struct;
};

struct object robot;
```