

Sessie 1: Basis

Gegeven het volgende C-programma:

```
int a, b, prod, aantal;  
  
main()  
{  
    aantal = getint();  
    a = 1;  
    while (a <= aantal)  
    {  
        b = 1;  
        while (b <= aantal)  
        {  
            prod = a * b;  
            printint (a, b, prod);  
            b += 1;  
        }  
        a += 1;  
    }  
}
```

- Leg in woorden uit wat dit programma doet (wees precies en beschrijf het programma alleen in termen van invoer - uitvoer)

Dit programma drukt alle mogelijk producten af van de getallen 1 t.e.m. een ingelezen getal; in de uitvoer worden zowel het getal, de vermenigvuldiger als het product afgedrukt.

- Vertaal dit programma naar DRAMA

```
        | a in R1  
        | b in R2  
        | prod in R3  
        | aantal in R4  
LEZ  
HIA    R4,R0    | aantal = getint()  
  
while:  HIA.w R1,1    | a = 1  
        VGL    R1,R4  
        VSP    GR,ewh | a <= aantal?  
  
while2: HIA.w R2,1    | b = 1  
        VGL    R2,R4  
        VSP    GR,ewh2 | b <= aantal?  
        HIA    R3,R1  
        VER    R3,R2    | prod = a * b  
  
        HIA    R0,R1  
        DRU                    | printint(a)  
        HIA    R0,R2  
        DRU                    | printint(b)  
        HIA    R0,R3  
        DRU                    | printint(prod)  
  
        OPT.w R2,1    | b += 1  
        SPR    while2  
  
ewh2:   OPT.w R1,1    | a += 1  
        SPR    while
```

ewh: STP
 EINDPR

Sessie 2: Rijen en Matrices

2e huistaak oefening. Lineariseer het volgende C-programma (matrix wordt kolom-per-kolom in het geheugen geplaatst):

```
int m[10][10];
int i, j, tmp;
main ()
{
    // inlezen van de matrix
    for (i=0; i<10; i++)
        for (j=0; j<10; j++)
            m[i][j] = getint();

    // voer complexe bewerking uit op de matrix
    for (i=0; i<10; i++)
        for (j=5; j<15; j++)
            m[i][j-5] += m[9-i][14-j];

    ... // Druk de matrix af
}
```

Oplossing:

```
int M[10*10];
int i, j, tmp;
int k, h;
main ()
{
    // inlezen van de matrix
    for (i=0; i<10; i++)
        for (j=0; j<10; j++) {
            k = 10 * j + i;
            M[k] = getint();
        }

    // voer complexe bewerking uit op de matrix
    for (i=0; i<10; i++)
        for (j=5; j<15; j++) {
            k = 10 * j + i - 50; // 10*(j-5) + i = 10*j + i - 50
            h = 149 - 10 * j - i // 10*(14-j) + (9-i) =
                                // 140 - 10*j + 9 - i = 149 - 10*j - i
            M[k] += M[h];
        }

    ... // Druk de matrix af
}
```

Of geoptimaliseerd (zonder vermenigvuldigingen):

```
int M[10*10];
int i, j, tmp;
int k, h;
main ()
```

```

{
    // inlezen van de matrix
    for (i=0; i<10; i++) {
        k = i; // (k met 1e waarde van j (=0))
        for (j=0; j<10; j++) {
            // k = 10 * j + i ==> telkens verhoogt k met 10
            M[k] = getint();
            k += 10;
        }
    }

    // voer complexe bewerking uit op de matrix
    for (i=0; i<10; i++) {
        k = i; // 10 * 5 + i - 50 (k met 1e waarde van j (=5))
        h = 99 - i; // 149 - 10 * 5 - i (h met 1e waarde van j (=5))
        for (j=5; j<15; j++) {
            // k = 10 * j + i - 50 ==> telkens verhoogt k met 10
            // h = 149 - 10 * j - i ==> telkens verlaagt h met 10
            M[k] += M[h];
            k += 10;
            h -= 10;
        }

        ... // Druk de matrix af
    }
}

```

Sessie 3: Functies I

3e huistaak oefening. Wat is de vertaling van het volgende C-programma:

```

void max(int n, int *m)
{
    register int i, getal;

    *m = 0;
    for (i=0; i<n; i++)
    {
        getal = getint();
        if (getal > *m)
            *m = getal;
    }
}

main ()
{
    int maxgetal;

    max(10, &maxgetal);
    printint(maxgetal);
}

```

Oplossing:

- Wat doet dit programma?
Leest 10 getallen in en drukt het maximum van deze rij af of 0 indien alle ingelezen getallen negatief zijn.
- ARs en toewijzingstabellen:

AR van max:

```
+-----+
|   TKA   |
+-----+
| oude R8 | <-- R8
+-----+
|    m    |
+-----+
|    n    |
+-----+
```

In registers:

Parameters: ---
Variabelen: i, getal

Toewijzingstabel (som)

Parameter	Adres
n	2(R8)
m	1(R8)

Variabele	Adres
i	R6
getal	R5

AR van main:

In registers: ---

```
+-----+
| maxgetal |
+-----+
| TKA = -1 |
+-----+
| oude R8 = -1 | <-- R8
+-----+
```

Toewijzingstabel (main)

Variabele	Adres
maxgetal	-2(R8)

- Vertaling:

```
| hoofdprogramma
main: HIA.w R0,-1
      BST   R0      | vorige R8
      HIA   R8,R9   | omg(main)
      BST   R0      | fake TKA

      | int maxgetal;
      AFT.w R9,1     | plaats voor maxgetal op de stapel

      | max(10, &maxgetal);

      | R0, R1 en R2 niet in gebruik
      | dus niets te bewaren

      HIA.w R0,10
      BST   R0      | par. n = 10
      HIA.a R0,-2(R8)
      BST   R0      | par. m = &maxgetal

      BST   R8      | bewaar omg(main)
```



```

|   TKA = -1   |   |
+-----+
| oude R8 = -1 | <---+
+-----+

```

Sessie 4: Functies II (Rijen)

2e huistaak oefening. Wat is de vertaling van het volgende C-programma?

```

int selsom (register int a[], register int n)
{
    register int som, i;
    int lim;

    som = 0;
    lim = n/3;

    for (i=1; i < lim; i += 3)
        som += a[3 * i - 2];

    return (som);
}

main()
{
    int rij[5];
    ...
    printint (selsom (rij, 3));
    ...
}

```

Oplossing:

- ARs en toewijzingstabellen:

AR van som:

```

+-----+
|   lim   |
+-----+
|   TKA   |
+-----+
| oude R8 | <--- R8
+-----+

```

In registers:

Parameters: a, n

Variabelen: som, i

Toewijzingstabel (som)

Parameter	Adres
a	R1
n	R2

Parameter	Adres
a	R1
n	R2

Variabele	Adres
som	R6
i	R5
lim	-2(R8)

Variabele	Adres
som	R6
i	R5
lim	-2(R8)

Variabele	Adres
som	R6
i	R5
lim	-2(R8)

Variabele	Adres
som	R6
i	R5
lim	-2(R8)

Variabele	Adres
som	R6
i	R5
lim	-2(R8)

AR van main:

```

+-----+

```

In registers: ---

```

| rij[0] |
+-----+
| rij[1] |
+-----+
| rij[2] |
+-----+
| rij[3] |
+-----+
| rij[4] |
+-----+
| TKA = -1 |
+-----+
| oude R8 = -1 | <-- R8
+-----+

```

Toewijzingstabel (main)

Variabele	Adres
rij	-2-4(R8) = -6(R8)

- Vertaling:

```

| hoofdprogramma
main: HIA.w R0,-1
      BST   R0      | vorige R8
      HIA   R8,R9   | omg(main)
      BST   R0      | fake TKA

      | int rij[5];
      AFT.w R9,5    | plaats op stapel voor rij

      | printint(selsom(rij, 10))

      | R0 niet in gebruik, R1 en R2 gebruikt voor parameters
      HIA.a R1,-6(R8) | par. a = &rij[0]
      HIA.w R2,10     | par. n = 10

      BST   R8      | bewaar omg(main)
      HIA   R8,R9   | omg(selsom)
      SBR   gsom     | selsom(rij, 10)
RET:  HST   R8      | herstel omg(main)

      | resultaat selsom in R0

      DRU                      | printint(selsom(...))
      ...

      STP

| functie selsom
      | register int som, i;
      | int lim;
selsom: AFT.w R9,1      | plaats voor lim op de stapel

      BST   R5      | bewaren omdat hierin i bijgehouden wordt
      BST   R6      | bewaren omdat hierin som bijgehouden wordt

      | lim = n/3
      HIA   R0,R1
      DEL.w R0,3

```

```

BIG    R0, -2(R8)

HIA.w  R6, 0      | som = 0
HIA.w  R5, 1      | i = 1
for:   VGL    R6, -2(R8) | i < lim?
      VSP    GRG, efor
      HIA    R0, R5
      VER.w  R0, 3
      AFT.w  R0, 2      | 3*i - 2
      OPT    R0, R1      | 3*i - 2 + &a[0] = &a[3*i - 2]
      OPT    R6, 0(R0)   | som += a[3*i - 2]
      OPT.w  R5, 1
      SPR    for
efor:   HIA    R0, R6      | resultaat (som) in R0
      HST    R6
      HST    R5          | registers herstellen
      OPT.w  R9, 1      | lim verwijderen van stapel
      KTG

EINDPR

```

- Volledige stapel

```

+-----+
| Kopie R6|   |
+-----+
| Kopie R5|   |
+-----+
| lim      |   | AR(selsom)
+-----+
| TKA = RET | <-- R9
+-----+
| oude R8   | <-- R8
+-----+
==+=====+==+=====+
| rij[0]    |   | <-- R1
+-----+
| rij[1]    |   |
+-----+
| rij[2]    |   |
+-----+
| rij[3]    |   |
+-----+
| rij[4]    |   |
+-----+
| TKA = -1  |   |
+-----+
| oude R8 = -1 | <-- AR(main)
+-----+

```

Sessie 5: Records

2e huistaak oefening. Wat doet dit C-programma? Wat is de vertaling van het volgende C-programma?

```

struct rationaalgetal
{
    int teller;
    int noemer;

```



```

    int gehdeel;
};

void leesrationaal(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;

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

```

Oplossing:

- Wat doet dit programma?

Het programma leest opeenvolgende rationale getallen in (bestaande uit: geheel-deel, teller, noemer) en drukt die af tot het getal oneindig ingelezen is (d.i. noemer = 0)

ARs en toewijzingstabellen

AR van leesrationaal:	In registers:
+-----+	Parameters: ---
TKA	Variabelen: ---
+-----+	
oude R8 <-- R8	
+-----+	
g	
+-----+	

Toewijzingstabel (leesrationaal)

Parameter		Adres

g		1(R8)

Variabele		Adres

---		---

AR van drukrationaal:	In registers:
+-----+	Parameters: ---
TKA	Variabelen: ---
+-----+	
oude R8 <-- R8	
+-----+	
g.teller	

```

+-----+
| g.noemer |
+-----+
| g.gehdeel |
+-----+

```

Toewijzingstabel (drukrationaal)

```

Parameter | Adres
-----+-----
g          | 1(R8)
-----+-----
Variabele | Adres
-----+-----
---       | ---

```

AR van main:

In registers: ---

```

+-----+
| q.teller |
+-----+
| q.noemer |
+-----+
| q.gehdeel |
+-----+
| TKA = -1 |
+-----+
| oude R8 = -1 | <-- R8
+-----+

```

Toewijzingstabel (main)

```

Variabele | Adres
-----+-----
q          | -4(R8)

```

```

| struct rationaalgetal
| teller:      +0
| noemer:      +1
| geheeldeel:  +2

```

```

main:      HIA.w R1, -1
           BST   R0      | omg(main)
           HIA   R8, R9
           BST   R0      | onecht TKA

           AFT.w R9, 3    | plaats voor var. q

           HIA.a R0, -4(R8)
           BST   R0      | parameter g = &q

           BST   R8
           HIA   R8, R9   | omg(leesrationaal)
           SBR   leesrationaal | leesrationaal(&q)
           HST   R8      | herstel omg(main)

           OPT.w R9, 1    | parameter weg

           HIA.w R0, 0
WHILE:     VGL   R0, -4+1(R8) | q.noemer != 0?
           VSP   GEL, EWHILE

```

```

| parameter q op de stapel
HIA R1, -4+2(R8) | q.gehdeel
BST R1
HIA R1, -4+1(R8) | q.noemer
BST R1
HIA R1, -4+0(R8) | q.teller
BST R1

BST R8
HIA R8, R9 | omg(drukrationaal)
SBR drukrationaal | drukrationaal(q)
HST R8 | herstel omg(main)

OPT.w R9, 3 | parameter weg

HIA.a R0, -4(R8)
BST R0 | parameter g = &q

BST R8
HIA R8, R9 | omg(leesrationaal)
SBR leesrationaal | leesrationaal(&q)
HST R8 | herstel omg(main)

OPT.w R9, 1 | parameter weg

SPR WHILE
EWHILE: STP

LEESRATIONAAL:
HIA R1, 1(R8) | R1 <-- g
LEZ
BIG R0, 2(R1) | g->gehdeel = getint()
LEZ
BIG R0, 0(R1) | g->teller = getint()
LEZ
BIG R0, 1(R1) | g->noemer = getint()
KTG

DRUKRATIONAAL:
HIA R0, 3(R8) | g.gehdeel
DRU
HIA R0, 1(R8) | g.teller
DRU
HIA R0, 2(R8) | g.noemer
DRU
KTG

EINDPR

```