

Oefenzitting 9 (24 november / 1 december)

Funcities V - Lijsten

1. Vertaal de procedure voegtoe en het hoofdprogramma uit de huistaak.

```
struct knoop
{
    struct knoop * links;
    struct knoop * rechts;
    int waarde;
};

void voegtoe (struct knoop ** bw, int g)
{
    if (*bw == NULL) {
        // maak een boom met 1 knoop
        *bw = alloc (sizeof(struct knoop));
        (*bw)->waarde = g;
        (*bw)->links = NULL;
        (*bw)->rechts = NULL;
    } else {
        if ((*bw)->waarde > g)
            // voeg toe in de linker tak indien het getal kleiner is dan huidige knoop
            voegtoe(&(*bw)->links, g);
        else
            // anders voeg je rechts toe
            voegtoe(&(*bw)->rechts, g);
    }
}

void druk_infix (register struct knoop * b)
{
    if (b != NULL) {
        druk_infix (b->links);
        printint(b->waarde);
        druk_infix (b->rechts);
    }
}

main ()
{
    register int getal;
    struct knoop * gb; // gesorteerde boom

    gb = NULL;
    getal = getint();
    while (getal != 0) {
        voegtoe (&gb, getal);
        getal = getint();
    }
}
```

```
    druk_infix(gb);  
}
```

2. Hoeveel geheugenplaatsen heb je nodig voor de matrix **robots**?

```
struct positie {  
    int x, y;  
    int hoogte;  
};  
  
struct tijdstip {  
    int dag, maand, jaar;  
    int uur, min;  
};  
  
struct reservatie {  
    struct tijdstip van, * tot;  
    struct werktuig (* lijst)[20];  
    struct positie afhalen; * terugbrengen;  
};  
  
struct werktuig {  
    int bezet;  
    int type;  
    struct werktuig * onderdelen[5];  
};  
  
struct robot {  
    struct positie huidig;  
    struct positie route [10];  
    struct reservatie werk;  
};  
  
struct robot robots[15][25];
```

Macro's

1. Schrijf een macro SOM die twee getallen optelt en het resultaat wegbergt.

```
| Macro SOM genereert code voor het optellen van 2 getallen (m.b.v. register Reg)
| deze getallen kunnen in het geheugen staan, in een register of een constante zijn
| daarom zullen de parameters AType en BType aangeven welk type de parameter heeft
| AType/BType == 0 ==> A/B staat in het geheugen (parameter = adres) of in een
|                               register (parameter = register)
| AType/BType == 1 ==> A/B is een constante
| Het resultaat moet in het geheugen weggeborgen worden (RType == 0) of in een
| register (Rtype == 1)
MACRO
    SOM ResType, Res, Atype, A, BType, B, Reg
    ...
MCREINDE
```

o Bijvoorbeeld:

```
SOM 0, RES, 0, A+3, 1, 15, R1
```

en

```
SOM 1, R3, 1, 4000, 0, B-7(R5), R0
```

moeten expanderen tot:

```
HIA    R1,A+3
OPT.w  R1,15
BIG    R1,RES
```

resp.

```
HIA.w  R0,4000
OPT    R0,B-7(R5)
HIA    R3,R0
```

o Controleer of de constante in het interval $[-5000, 4999]$ ligt! Anders geef je een fout.

o Expandeer met je macro de volgende macro-oproepen:

```
SOM 0, 3(R8), 0, -2(R8), 0, R5, R2
SOM 1, R4, 1, -3000, 1, 2000, R0
SOM 0, RES, 1, 8000, 0, -6(R8), R6
```

o Pas je macro aan zodat bij constanten buiten het interval $[-5000, 4999]$, de constante in het geheugen geplaatst wordt.

Schrijf hiervoor een aparte macro (CONST) die je vanuit SOM oproept.

Opgelet! Je moet er wel voor zorgen dat het programma over deze constante springt!

De gegenereerde code voor de derde SOM-oproep hierboven zou het volgende moeten zijn (de eerste 2 lijnen worden door CONST gegenereerd, en de rest door SOM):

```
    SPR _3HIA
    8000
_3HIA: HIA    R6,_3HIA - 1
      OPT    R6,-6(R8)
      BIG    R6,RES
```

Vergeet niet de label (hier _3HIA) als parameter door te geven aan CONST!

2. Welke code wordt er door de voorvertaler gegenereerd?

```
MEVA  N,7

      HIA.w R0,0
      MEVA TELLER,<N>
$LUS: MNTS
      OPT.w R0,<TELLER>
      MEVA TELLER,<TELLER>-1
      MVGL TELLER,1
      MVGL GR,$LUS
      DRU
      MSPR $END
$TRG: MNTS

      HIA.w R0,0
      HIA.w R1,<N>
LUS:  VGL.w R1,1
      VSP   KLG,ELUS
      OPT   R0,R1
      SPR   LUS
ELUS: DRU
$END: MEVA TELLER,<TELLER>-1
      MVGL TELLER,0
      MSPR GRG,$TRG

      MEVA TELLER,<N>
      MEVA SOM,0
$WH:  MEVA SOM,<SOM>+<TELLER>
      MEVA TELLER,<TELLER>-1
      MVGL TELLER,1
      MVSP GR,$WH

      HIA.w R0,<SOM>
      DRU
EINDPR
```

- ☐ Wat is het resultaat van de uitvoering van dit programma?
- ☐ Wat is de grootste waarde die de voorvertaler-variabele "N" mag aannemen zodat het programma nog correct vertaald kan worden?