

Huistaak - Functies V (Lijsten)

Oefening 1

Gegeven het volgende C-programma:

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

void voegtoe (struct knoop ** bw, register int g) {
    if (*bw == NULL) {
        // maak een boom met 1 knoop
        // sizeof(...) is geen functieoproep maar een directief voor de
        // compiler! Namelijk: bepaal de grootte van dit type
        *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 de
            // 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);
}

```

Voor dit C-programma:

- Er wordt gewerkt met een binaire boom, waarbij de ingelezen getallen gesorteerd worden bijgehouden. Probeer eerst de C-code te begrijpen.
- Teken de activatierecords voor **druk_infix**, **voegtoe** en **main**, en stel de toewijzingstabellen op
- Vertaal de procedure **druk_infix** (de procedure **voegtoe** en het hoofdprogramma worden tijdens de oefenzitting vertaald).
- Teken hoe de boom eruit ziet als je de volgende rij getallen hebt ingelezen (in die volgorde): 8, 3, 7, 13, 9, 11, 2, 6, 15, 10.
 - Hoeveel (recursieve) oproepen van voegtoe zullen er uitgevoerd worden wanneer we aan deze boom het getal 13 moeten toevoegen?
 - Teken de volledige stapel zoals ze eruit ziet als de knoop wordt gealloceerd (net voor de inialisatie).

Oefening 2

Welke DRAMA-code genereert de voorvertaler?

```
MEVA <DEBUG>,5
```

```
MACRO
```

```
    TRANSPORT op,par
    <op> R1,<par>
```

```
MCREINDE
```

```
MACRO
```

```
    ABSWSOM A,IDX,Reg
    HIA    <Reg>,<A>+<IDX>
    MEVA   <IDX2>,<IDX>+1
    OPT    <Reg>,<A>+<IDX2>
    MEVA   <IDX2>,<IDX>-1
    VSP    NNEG,$POS
    VER.w  <Reg>,-1
```

```
$POS:  BIG    <Reg>,<A>+<IDX2>
```

```
MCREINDE
```

```
    HIA.w  R3,100
```

```
    TRANSPORT HIA.w,2(R3)
```

```
    TRANSPORT BIG.i,A+3
```

```
    TRANSPORT VGL.w,5
```

```
HIER:  VSP    GR,HIER+2
```

```
    TRANSPORT OPT,A-7(R3)
```

```
    MVGL   <DEBUG>,0
```

```
    MVSP   GRG,$NXT
```

```
    ABSWSOM A,5,R1
```

```
    MSPR   $EINDE
```

```
$NXT:  MNTS
```

```
    ABSWSOM A,10,R2
```

```
$EINDE: MNTS
```

```
A: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

```
EINDPR
```