

Oefenzitting 8 - Lijsten - versie 2

Huistaak

1. Wat is de vertaling van het volgende C-programma?

```
int * iw;

main()
{
    iw = alloc(1);
    printint (iw, *iw);
    *iw = 10;
    printint (iw, *iw);
    iw = (int *) 10;

    // verklaring van (int *):
    // d.i. een type-conversie
    // d.w.z. dat 10 als een adres moet beschouwd worden i.p.v. als een integer
    // zonder deze conversie geeft de compiler een waarschuwing of een fout
    // voor de vertaling naar DRAMA heeft dit geen enkel belang
    // behalve wanneer je een adres zou willen gebruiken dat groter is dan 4999
    // Bijv.   iw = (int *) 8999; kan vertaald worden als: HIA.a Ri,8999 ...
    //        iw = 8999;   kan niet meer vertaald worden met een directe operand

    printint (iw, *iw);
}
```

Voer dit programma uit met de simulator en verklaar de 6 getallen die afgedrukt worden.

2. Wat is de vertaling van het volgende C-programma? Het programma leest een aantal getallen in en bouwt een gesorteerde lijst op (van klein naar groot). De lijst heeft een kopelement.

```
struct elem
{
    int info;
    struct elem * volg;
};

void inlassen (struct elem * l, int g)
{
    register struct elem * p, *pp;
    struct elem * n;

    pp = l;
    p = pp->volg;
    while ((p != NULL) && (p->info < g))
    {
        pp = p;
        p = pp->volg;
    }
    // Het nieuwe element toevoegen tussen pp en p
    n = alloc(sizeof(struct elem)); // sizeof(...) geeft de grootte van het type
                                    // dit is GEEN functieoproep maar een opdracht voor de compiler!
    n->info = g;
    n->volg = p;
    pp->volg = n;
}

void druklijst (register struct elem * l)
{
    l = l->volg; // sla kopelement over
    while (l != NULL) {
        printint(l->info);
        l = l->volg;
    }
}

main ()
{
    register struct elem * lst;
    int getal;

    // maak lege lijst (met kopelement)
    lst = alloc(2);
    lst->info = 0;
    lst->volg = NULL;

    // inlezen van positieve getallen
    getal = getint();
    while (getal > 0)
    {
        inlassen(lst, getal);
        getal = getint();
    }
    druklijst(lst);
}
```

3. Vraag 2 van de eerste test

```

int data[] = {1,2,3,4,5};

int bereken(int * ptr, register int n)
{
    int tmp[3];
    register int k;

    ...
    if (n >= 3)
        return (0);
    else
        return (bereken(&tmp[k-1], n+1) + *ptr - n);
}

main ()
{
    int i;
    register int j;

    j = 1;
    i = bereken (&data[j-1], j);
    printint (data[j-i+2]);
}

```

- o Teken de ARs en toewijzingstabellen voor main en bereken
- o Vertaal de procedure bereken en main
- o Teken de volledige stapel zoals ze er uit ziet op het ogenblik dat de 2de oproep van bereken (d.i. de 1ste recursieve oproep) gekomen is aan de SBR bereken (d.i. de 2de recursieve oproep)
Dus: main --> bereken --> bereken (juist voor de nieuwe oproep: "SBR bereken")