

Huiswerk - Sander & Joran

Opgave 13

1.
stack: NILL 0
Volgende: num(2)
Rest: +x*3
Actie: s5
2.
stack: NILL 0 num(2) 5
Volgende: ADD
Rest: x*3
Actie: r9
3.
stack: NILL 0 Factor 3
Volgende: ADD
Rest: x*3
Actie: r7
4.
stack: NILL 0 Term 2
Volgende: ADD
Rest: x*3
Actie: r4
5.
stack: NILL 0 Expr 1
Volgende: ADD
Rest: x*3
Actie: s7
6.
stack: NILL 0 Expr 1 ADD 7
Volgende: ID(x)
Rest: *3
Actie: s6
7.
stack: NILL 0 Expr 1 ADD 7 ID(x) 6
Volgende: MUL
Rest: 3
Actie: r10
8.
stack: NILL 0 Expr 1 ADD 7 Factor 3
Volgende: MUL
Rest: 3
Actie: r7
9.
stack: NILL 0 Expr 1 ADD 7 Term 18
Volgende: MUL

Rest: 3
Actie: s9

10. **stack:** NILL 0 Expr 1 ADD 7 Term 18 mul 9
Volgende: 3
 Rest:
 Actie: s5

11. **stack:** NILL 0 Expr 1 ADD 7 Term 18 mul 9 num(6) 5
Volgende:
 Rest:
 Actie: r9

12. **stack:** NILL 0 Expr 1 ADD 7 Term 18 mul 9 Factor 3
Volgende:
 Rest:
 Actie: r5

13. **stack:** NILL 0 Expr 1 ADD 7 Term 18
Volgende:
 Rest:
 Actie: r3

14. **stack:** NILL 0 Expr 1
Volgende:
 Rest:
 Actie: Accept

Opgave 14

```
extern void printBool();

int age = 27;
int thresh = 40;

export void main()
{
    printIfOld(isOld(age, thresh));
}

void printIfOld(bool old)
{
    // 4: The local variable old shadows printIfOld()'s parameter old.
    bool old;

    // 6: The local variable age is initialised by the global variable age.
    int age = age;

    age = 2 * age;

    old = isOld(age, thresh);

    printBool(old);
}

// 3: isOld()'s parameter age shadows the global variable age.
bool isOld(int age, int thresh)
{
    // 7: The local variable printIfOld shadows the function printIfOld().
    bool printIfOld = true;

    // 1: The local variable diff never leaves this scope, thus is non-escaped.
    int diff;

    // 2: The local variable thr is used by subfunction thresh(), thus escaped.
    int thr = 25;

    // 5: The local function thresh() shadows isOld()'s parameter thresh.
    int thresh()
    {
        return thr;
    }

    diff = age - thresh();

    return (diff > 0);
}
```