Code Generation templates using an intermediary stack machine

```
        Push k:
                DATA #LIT + lit k
                LOAD R8,DR
                STORE R8,R9
                INC R9,-1

        Get x:
                DATA #VAR + x
                LOAD R8,DR
                STORE R8,R9
                INC R9,-1

        Put x:
                DATA #VAR + x
                INC R9,1
                LOAD R8,R9
                STORE R8,DR
                INC R9,-1

        Pop:
                INC R9,1

        DoOp Plus:
                INC R9,1
                LOAD R8,R9
                INC R9,1
                LOAD R7,R9
                ADD R7,R8
                STORE R8,R9
                INC R9,-1

        DoOp Diff:
                INC R9,1
                LOAD R8,R9
                INC R9,1
                LOAD R7,R9
                SUB R7,R8
                STORE R8,R9
                INC R9,-1

        Input:
                READ R8,DD
                STORE R8,R9
                INC R9,-1
```

```
DoOp Mult:
        ZERO R4
        INC R9,1
        LOAD R0,R9
        INC R9,1
        LOAD R1,R9
        COPY R0,R2
        COPY R1,R3
        SHIFT R2, 5
        SHIFT R2, 6
        SHIFT R3, 5
        SHIFT R3, 6
        SUB R3,R2
        SHIFT R0,-1
        SHIFT R0, 1
        SHIFT R1,-1
        SHIFT R1, 1
        DATA #SKIP + i
        COPY DR,J3
        DATA #LOOP + i
        COPY DR,J2
        #LOOP + i
        SET R3, 1
        AND R1,R3
        JPIF R3,EZ,J3
        ADD R0,R4
        #SKIP + i
        SHIFT R0,-1
        SHIFT R1, 1
        JPIF R1,NZ,J2
        DATA #DONE + i
        COPY DR,J3
        JPIF R2,EZ,J3
        ZERO R3
        SUB R4,R3
        COPY R3,R4
        #DONE + i
        STORE R4,R9
        INC R9,-1
```