```
%  Assignment 4 Sample Program 2                    % 2-1
{                                                   % 2-2
   var a, b, c , d : integer                        % 2-3
   var p, q, r : boolean                            % 2-4
   PUSHMT
   SETD 0
   PUSH UNDEFINED
   PUSH main_needed_words
   DUPN
   a := b + c - d*c + b/c                           % 2-5
   ADDR (LL ON of a)
   ADDR (LL ON of b)
   LOAD
   ADDR (LL ON of c)
   LOAD
   ADD
   ADDR (LL ON of d)
   LOAD
   ADDR (LL ON of c)
   LOAD
   MUL
   SUB
   ADDR (LL ON of b)
   LOAD
   ADDR (LL ON of c)
   LOAD
   DIV
   ADD
   STORE
   p := not false or not (q and r)                  % 2-6
   PUSH(0)
   PUSH(1)
   SUB
   NEG // not
   ADDR (LL ON of p)
   ADDR (LL ON of q)
   LOAD
   ADDR (LL ON of r)
   LOAD
   MUL // and
   PUSH(1)
   SUB
   NEG // not
```

```
OR
STORE
if p then  a := 3 fi                          % 2-7
ADDR (LL ON of p)
LOAD
PUSH(2-8)
BF
ADDR (LL ON of a)
PUSH(3)
STORE
if p or not p then b := 2 else b := 0 fi      % 2-8
ADDR (LL ON of p)
LOAD
ADDR (LL ON of p)
LOAD
PUSH(1)
SUB
NEG
OR
PUSH(else_case)
BF

ADDR (LL ON of b)
PUSH(2)
STORE
PUSH(2-9)
BR

else_case:
ADDR (LL ON of b)
PUSH(0)
STORE
while c < 7 do  c := 6  end                   % 2-9
ADDR (LL ON of c)
LOAD
PUSH(7)
LT
PUSH(2-10)
BR
ADDR (LL ON of c)
PUSH(6)
STORE
PUSH(2-9)
```

```
BR
while true do  b := b + 1  end                    % 2-10
PUSH(1)
PUSH(2-11)
BF
ADDR (LL ON of b)
ADDR (LL ON of b)
LOAD
PUSH(1)
STORE
PUSH(2-10)
BR
repeat { a := 3 exit b := 7 } until false         % 2-11
ADDR (LL ON of a)
PUSH(3)
STORE
PUSH(2-12)
BR
ADDR (LL ON of b)
PUSH(7)
STORE
PUSH(0) // false
PUSH(2-11)
BF
while (q or (r and (not p))) do  exit when b not= 10 end    % 2-12
ADDR (LL ON of q)
LOAD
ADDR (LL ON of r)
LOAD
ADDR (LL ON of p)
PUSH(1)
SUB
NEG
MUL
OR
PUSH(2-13)
BF
ADDR (LL ON of b)
LOAD
PUSH(10)
EQ
PUSH(1)
SUB
```

```
NEG
PUSH(1) // exit when start
SUB
NEG
PUSH(2-13)
BF // exit when end

PUSH(2-12)
BR
put "Value is " , a / b, " or not ", newline     % 2-13
PUSH('V')
PRINTC
… // PUSH and PRINTC every char
ADDR (LL ON of a)
ADDR (LL ON of b)
DIV
PRINTI
PUSH(' ')
PRINTC
… // PUSH and PRINTC every char
PUSH('\n')
PRINTC
while true not= false do  {                      % 2-14
PUSH(1)
PUSH(0)
EQ
PUSH(1)
SUB
NEG
PUSH(2-34)
BF
        var b1, b2 : boolean                     % 2-15
        get  a, c, b                             % 2-16
         ADDR (LL ON of a)
        READI
        STORE
        ADDR (LL ON of c)
        READI
        STORE
        ADDR (LL ON of b)
        READI
        STORE
        exit when p or not r                     % 2-17
```

```
ADDR(LL ON of p)
LOAD
ADDR (LL ON of r)
LOAD
PUSH(1)
SUB
NEG
OR
PUSH(1) // exit when start
SUB
NEG
PUSH(2-34)
BF // exit when end
b1 := not p or q                        % 2-18
ADDR (LL ON of b1)
ADDR (LL ON of p)
LOAD
PUSH(1)
SUB
NEG
ADDR (LL ON of q)
LOAD
OR
STORE
repeat  {                               % 2-19
var w, x , A[100] : integer             % 2-20
p := ( b2 ? q : d <= 7)                 % 2-21
ADDR (LL ON of p)
ADDR (LL ON of b2)
PUSH(false_case)
BF
ADDR (LL ON of q)
LOAD
PUSH(store_stmt)
BR
false_case:
ADDR (LL ON of d)
LOAD
PUSH(7)
SWAP
LT
PUSH(1)
SUB
```

```
NEG
store_stmt:
STORE
if w <= a then exit fi              % 2-22
ADDR (LL ON of w)
LOAD
ADDR (LL ON of a)
LOAD
SWAP
LT
PUSH(1)
SUB
NEG
PUSH(2-23)
BF
PUSH(2-34)
BR
while  p or not q or r do                % 2-23
ADDR (LL ON of p)
LOAD
ADDR (LL ON of q)
LOAD
PUSH(1)
SUB
NEG
OR
ADDR (LL ON of r)
LOAD
OR
PUSH(2-30)
BF
{                                        % 2-24
        var  t, u : integer              % 2-25
        p := not q                       % 2-26
        ADDR (LL ON of p)
        ADDR (LL ON of q)
        LOAD
        PUSH(1)
        SUB
        NEG
        STORE
        t := ( p or q ? t + 1 : t - 1 )  % 2-27
        ADDR (LL ON of t)
```

```
        ADDR (LL ON of p)
        LOAD
        ADDR (LL ON of q)
        LOAD
        OR
        PUSH(false_case)
        BF
        ADDR (LL ON of t)
        LOAD
        PUSH(1)
        ADD
        PUSH(store_stmt)
        BR
        false_case:
        ADDR (LL ON of t)
        PUSH(1)
        SUB
        store_stmt:
        STORE
exit  when t > 12                % 2-28
ADDR (LL ON of t)
LOAD
PUSH(12)
SWAP
LT
PUSH(1) // exit when start
SUB
NEG
PUSH(2-30)
BF // exit when end
}                                % 2-29
PUSH(2-23)
BR
end   % while                    % 2-30
exit when A[w] < d               % 2-31
ADDR (LL ON of A)
ADDR (LL ON of w)
LOAD
PUSH(lower bound of A)
SUB // calculate the correct offset
ADD
LOAD
ADDR(LL ON of d)
```

```
        LOAD
        LT
        PUSH(1) // exit when start
        SUB
        NEG
        PUSH(2-33)
        BF // exit when end
        } until p and r  % repeat                % 2-32
        ADDR (LL ON of p)
        LOAD
        ADDR (LL ON of r)
        LOAD
        MUL
        PUSH(2-19)
        BF
  }                                              % 2-33
  PUSH(2-14)
  BR
  end  % while                                   % 2-34
}                                                % 2-35
```