```
% Assignment 4 Sample Program 3                          % 3-1
{                                                       % 3-2
  var a, b, c, d : integer                              % 3-3
  var p, q, r, s : boolean                              % 3-4
  PUSHMT
  SETD 0
  PUSH UNDEFINED
  PUSH main_needed_words
  DUPN
  forward proc Q (a:boolean, c:integer, d : boolean)    % 3-5
  proc P () {                                           % 3-6
        var e : integer                                 % 3-7
        while q do                                      % 3-8
        ADDR (LL ON of q)
        LOAD
        PUSH(3-11)
        BF
        Q( p , e + a - 1, c <= d )                      % 3-9
        PUSH UNDEFINED  // return value, to be filled in,
        PUSH(3-10)
        ADDR LL 0           // saved display reference

        // display update
        PUSHMT
        SETD LL             // LL of function / procedure

        //p
        ADDR (LL ON of p)
        LOAD

        //e+a-1
        ADDR (LL ON of e)
        LOAD
        ADDR (LL ON of a)
        LOAD
        ADD
        PUSH(1)
        SUB

        //c<=d
        ADDR (LL ON of c)
        LOAD
        ADDR (LL ON of d)
```

```
        LOAD
        SWAP
        LT
        PUSH(1)
        SUB
        NEG

        PUSH(3-17)
        BR

        return                                  % 3-10
        PUSH num_params + num_local_words
        POPN
        SETD LL                 // LL of function / procedure
        BR
        end                                     % 3-11
        }  % P                                          % 3-12
func F( m : integer, n : boolean ) : integer   % 3-13
        {                                               % 3-14
        result ( n ?  m + b : F( m - b , n and not s))    % 3-15

        ADDR LL 0
        PUSH 3
        SUB //push the address of the return value, which is the display base address - 3

        ADDR (LL ON of n)
        LOAD
        PUSH(false_case)
        BF
        ADDR (LL ON of m)
        LOAD
        ADDR (LL ON of b)
        LOAD
        ADD
        PUSH(result_case)
        BR

        fasle_case:
        // F( m - b , n and not s)
        PUSH UNDEFINED  // return value, to be filled in,
        PUSH(result_case)
        ADDR LL 0               // saved display reference
```

```
        // display update
        PUSHMT
        SETD LL

        ADDR (LL ON of m)
        LOAD
        ADDR (LL ON of b)
        LOAD
        SUB

        ADDR (LL ON of n)
        LOAD
        ADDR (LL ON of s)
        LOAD
        PUSH 1
        SUB
        NEG
        MUL

        PUSH(3-14) // function call F
        BR

        result_case:
        // done  evaluating result expression
        STORE

        PUSH num_params + num_local_words
        POPN
        SETD LL                 // LL of function / procedure
        BR

        }  % F                                          % 3-16
proc Q( m : boolean , n : integer , p : boolean)    % 3-17
        {                                               % 3-18
        var t, u, v  : integer                  % 3-19
        func G()  : integer                     % 3-20
        {                                       % 3-21
        var w, x : integer                      % 3-22
        Q( not m , a + u - x , p or s )         % 3-23
        PUSH UNDEFINED   // return value, to be filled in,
        PUSH(3-24)
        ADDR LL 0               // saved display reference
```

```
// display update
PUSHMT
SETD LL

// not m
ADDR (LL ON of m)
LOAD
PUSH(1)
SUB
NEG
//a+u-x
ADDR (LL ON of a)
LOAD
ADDR (LL ON of u)
LOAD
ADD
ADDR (LL ON of x)
LOAD
SUB
//p or s
ADDR (LL ON of p)
LOAD
ADDR (LL ON of s)
LOAD
OR

PUSH(3-17)
BR

result (  m or p ? v + n : u - b )          % 3-24

ADDR LL 0
PUSH 3
SUB //push the address of the return value, which is the display base address - 3

ADDR (LL ON of m)
LOAD
ADDR (LL ON of p)
LOAD
OR
PUSH(false_case)
BF
ADDR (LL ON of v)
```

```
LOAD
ADDR (LL ON of n)
LOAD
ADD
PUSH(result_case)
BR
ADDR (LL ON of u)
LOAD
ADDR (LL ON of b)
LOAD
SUB

result_case:
// done  evaluating result expression
STORE

PUSH num_params + num_local_words
POPN
SETD LL                 // LL of function / procedure
BR
}  % G                                  % 3-25
if G()  < 7 then return fi              % 3-26
PUSH UNDEFINED   // return value, to be filled in,
PUSH(compare_case)
ADDR LL 0               // saved display reference

// display update
PUSHMT
SETD LL

PUSH(3-20)
BR

compare_case:
PUSH(7)
LT
PUSH(3-27)
BF

// return
PUSH num_params + num_local_words
POPN
SETD LL                 // LL of function / procedure
```

```
        BR
        if F( t, not r ) = 17 then return fi          % 3-27
        //function_call of F()
        PUSH UNDEFINED   // return value, to be filled in,
        PUSH(compare_case)
        ADDR LL 0               // saved display reference

        ADDR (LL ON of t)
        LOAD
        ADDR (LL ON of r)
        LOAD
        PUSH(1)
        SUB
        NEG
        PUSH(3-13)
        BR

        compare_case:
        PUSH(17)
        EQ
        PUSH(3-28)
        BF

        // return
        PUSH num_params + num_local_words
        POPN
        SETD LL                 // LL of function / procedure
        BR
        P()                                           % 3-28
        PUSH UNDEFINED   // return value, to be filled in,
        PUSH(3-29)
        ADDR LL 0               // saved display reference

        PUSH(3-6)
        BR
} %Q                                                  % 3-29
Q( not p or q , b * c , p not= q )                    % 3-30
PUSH UNDEFINED // return value, to be filled in,
PUSH(3-31)
ADDR LL 0              // saved display reference

ADDR (LL ON of p)
LOAD
```

```
  PUSH(1)
  SUB
  NEG
  ADDR (LL ON of q)
  LOAD
  OR
  ADDR (LL ON of b)
  LOAD
  ADDR (LL ON of c)
  LOAD
  MUL
  ADDR (LL ON of p)
  LOAD
  ADDR (LL ON of q)
  LOAD
  EQ
  PUSH(1)
  SUB
  NEG

  PUSH(3-17)
  BR
}                                   % 3-31
```