



Run-time Environments

Lecture 10

Exercise

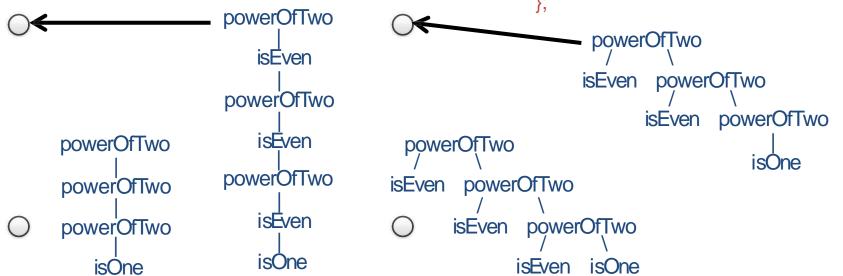
Question?

The powerOfTwo() function, shown to the right, returns true if its argument is a power of two, false otherwise. What is the activation tree for powerOfTwo(4)?

```
if isEven(x) then powerOfTwo(x / 2)
   else isOne(x)
 };
       powerOfTwo
      isEven
             powerOfTwo
            isEven powerOfTwo
powerOfTwo
```

isEven(x:Int): Bool $\{x\%2=0\}$;

 $isOne(x:Int) : Bool \{ x = 1 \};$ powerOfTwo(x:Int) : Bool {



Profs. Aiken

Answer!

The powerOfTwo() function, shown to the right, returns true if its argument is a power of two, false otherwise. What is the activation tree for powerOfTwo(4)?

```
else isOne(x)
                                                };
                powerOfTwo
                                                      powerOfTwo
                   isEven
                                                     isEven
                                                            powerOfTwo
                powerOfTwo
                                                           isEven
                                                                  powerOfTwo
                                  powerOfTwo
                   isEven
powerOfTwo
                powerOfTwo
                                 isEven
                                        powerOfTwo
powerOfTwo
                   isEven
powerOfTwo
                                       isEven
                                               powerOfTwo
                    isOne
   isOne
                                              isEven isOne
```

isEven(x:Int): Bool $\{x\%2=0\}$;

if isEven(x) then powerOfTwo(x / 2)

isOne(x:Int) : Bool { x = 1 }; powerOfTwo(x:Int) : Bool {

Question?

```
program main ()
   var a, b : int;
   procedure f (m : int);
        var v: int:
       procedure q (y : int);
           var i, j : int;
           f (y);
       end q;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
               var k : int:
              g (i - 1);
        \rightarrow b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f:
   f (4)
end main:
```

Assume that in running the given program, the procedures are invoked in the order that follows: f(4), p(3), q(2), g(1), and f(1).

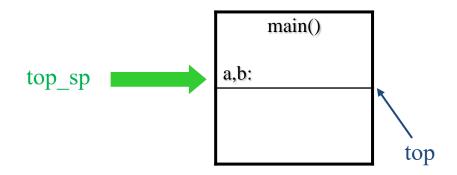
Draw the runtime stack at the end of these invocations. Assume that non-local variables are addressed using Access Links.

Assume there is any gap between access links and local data

What is the addresses of variables in assignment b := m + v in procedure q?

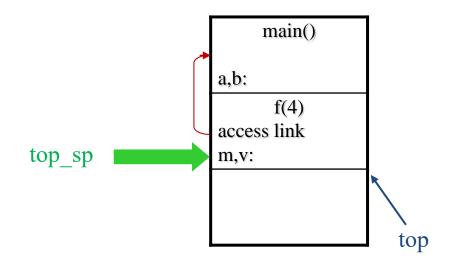
Answer!

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int;
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int;
              g (i - 1);
              b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f (4)
end main;
```



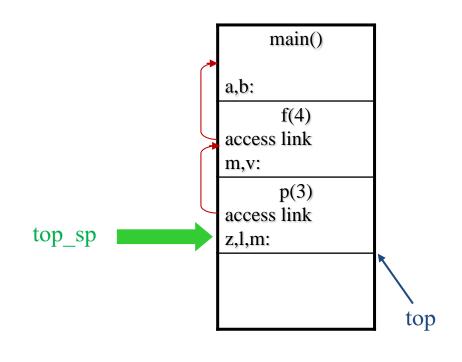
Answer! (cont.), when f(4) is called

```
program main ()
   var a, b : int;
  procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if(z > 4) f(z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f(4)
end main:
```



Answer! (cont.), when p(3) is called

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f (4)
end main:
```



Answer! (cont.), when q(2) is called

end main:

```
program main ()
   var a, b : int;
                                                                                     main()
   procedure f (m : int);
       var v: int:
                                                                                a.b:
       procedure g (y : int);
                                                                                      f(4)
           var i, j : int;
                                                                                access link
           f (y);
                                                                                m,v:
       end g;
                                                                                      p(3)
       procedure p (z : int);
                                                                                access link
           var I, m : int;
                                                                                z,1,m:
     procedure q (i : int);
                                                                                      q(2)
              var k : int:
                                                                                access link
              g (i - 1);
                                                                                i,k:
                                                         top sp
              b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
                                                                                                     top
       if m > 1 then p(3)
   end f;
   f(4)
```

Answer! (cont.), when g(1) is called

```
program main ()
   var a, b : int;
                                                                                      main()
   procedure f (m : int);
       var v: int:
                                                                                 a.b:
    → procedure q (y : int);
                                                                                       f(4)
           var i, j : int;
                                                                                 access link
           f (y);
                                                                                 m,v:
       end g;
                                                                                       p(3)
       procedure p (z : int);
                                                                                 access link
           var I, m : int;
                                                                                 z,1,m:
           procedure q (i : int);
                                                                                       q(2)
               var k : int:
                                                                                 access link
              g (i - 1);
                                                                                 i.k:
              b := m + v;
           end q;
                                                                                       g(1)
          if(z > 4) f(z) else q(z - 1)
                                                                                 access link
                                                          top sp
       end p;
                                                                                 y,i,j:
       if m > 1 then p(3)
   end f;
   f(4)
                                                                                                       top
end main:
```

Answer! (cont.), when f(1) is called

```
program main ()
   var a, b : int;
                                                                                       main()
procedure f (m : int);
        var v: int:
                                                                                  a.b:
        procedure g (y : int);
                                                                                        f(4)
           var i, j : int;
                                                                                  access link
           f (y);
                                                                                 m,v:
       end g;
                                                                                        p(3)
        procedure p (z : int);
                                                                                  access link
           var I, m : int;
                                                                                  z,1,m:
           procedure q (i : int);
                                                                                        q(2)
               var k : int:
                                                                                  access link
               g (i - 1);
                                                                                 i.k:
              b := m + v;
           end q;
                                                                                        g(1)
          if (z > 4) f (z) else a(z - 1)
                                                                                  access link
       end p;
                                                                                  y,i,j:
       if m > 1 then p(3)
                                                                                        f(1)
   end f;
                                                                                  access link
   f(4)
                                                            top sp
                                                                                  m,v:
end main:
                                                                                                 10
```

Answer! (cont.), when returned to g(1)

```
program main ()
   var a, b : int;
                                                                                       main()
   procedure f (m : int);
        var v: int:
                                                                                 a.b:
     procedure g (y : int);
                                                                                        f(4)
           var i, j : int;
                                                                                 access link
           f (y);
                                                                                 m,v:
       end g;
                                                                                        p(3)
        procedure p (z : int);
                                                                                 access link
           var I, m : int;
                                                                                 z,1,m:
           procedure q (i : int);
                                                                                        q(2)
               var k : int:
                                                                                 access link
               g (i - 1);
                                                                                 i.k:
              b := m + v;
           end q;
                                                                                        g(1)
          if (z > 4) f (z) else q(z - 1)
                                                                                 access link
                                                          top sp
       end p;
                                                                                 y,i,j:
       if m > 1 then p(3)
   end f;
   f (4)
                                                                                                       top
end main:
```

Answer! (cont.), when returned to q(2)

```
program main ()
   var a, b : int;
                                                                                 main()
   procedure f (m : int);
       var v: int:
                                                                            a.b:
       procedure q (y : int);
                                                                                   f(4)
           var i, j : int;
                                                                            access link
           f (y);
                                                                            m,v:
       end q;
                                                                                  p(3)
       procedure p(z:int);
                                                                            access link
          var I, m: int;
                                                                            z.l.m:
      ---> procedure q (i : int);
                                                                                  q(2)
              var k : int:
                                                                            access link
              g (i - 1);
                                                      top sp
                                                                            i.k:
             b := m + v:
          end q;
          if (z > 4) f (z) else a(z - 1)
       end p;
                                                                                                 top
      if m > 1 then p(3)
   end f:
                                 address m: @(top_sp - #1) + #1 + #2
   f(4)
end main:
                                 address v: @(@(top_sp - #1)) + #1 + #1
                                                                                           12
                                 _address b: @(@(@(top_sp - #1))) + #1 + #1
```

Question?

```
program main ()
   var a, b : int;
   procedure f (m : int);
        var v: int:
       procedure q (y : int);
           var i, j : int;
           f (y);
       end q;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
               var k : int:
              g (i - 1);
        \rightarrow b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f:
   f (4)
end main:
```

Assume that in running the given program, the procedures are invoked in the order that follows: f(4), p(3), q(2), g(1), and f(1).

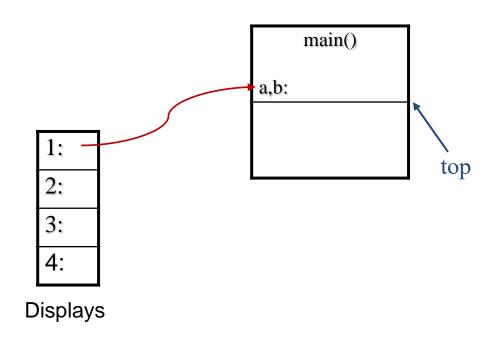
Draw the runtime stack at the end of these invocations. Assume that non-local variables are addressed using Displays.

Assume there is any gap between access links and local data

What is the addresses of variables in assignment b := m + v in procedure q?

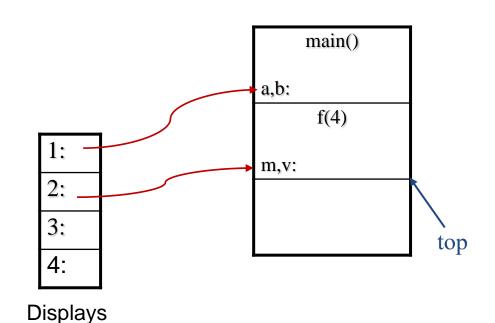
Answer!

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int;
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
               var k : int;
              g (i - 1);
              b := m + v;
           end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f (4)
end main;
```



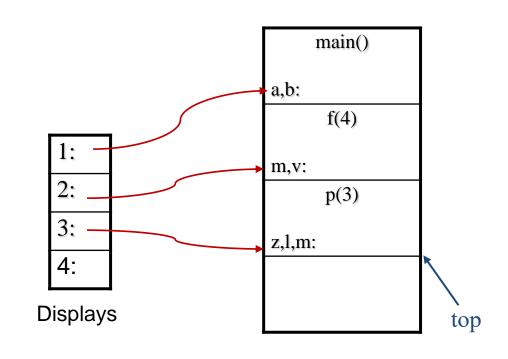
Answer! (cont.), when f(4) is called

```
program main ()
   var a, b : int;
  procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if(z > 4) f(z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f(4)
end main:
```



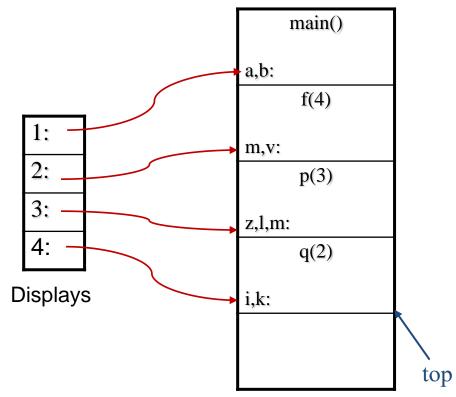
Answer! (cont.), when p(3) is called

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if(z > 4) f(z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f(4)
end main:
```



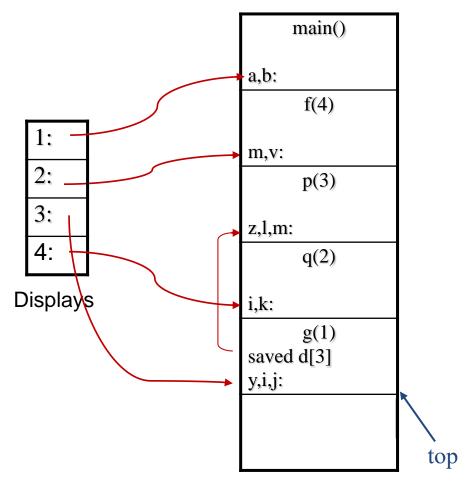
Answer! (cont.), when q(2) is called

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end q;
       procedure p (z : int);
           var I, m : int;
                                                      3:
       → procedure q (i : int);
              var k : int:
              g (i - 1);
                                                   Displays
             b := m + v;
           end q;
          if(z > 4) f(z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f(4)
end main:
```



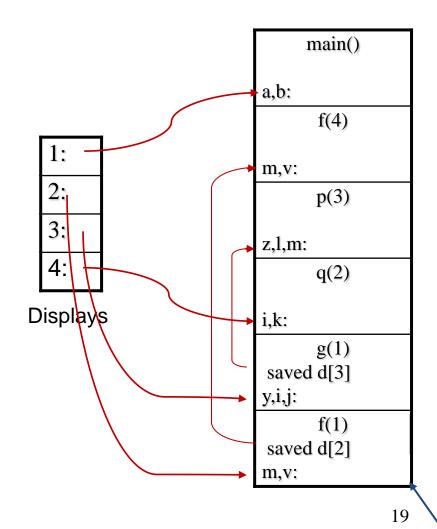
Answer! (cont.), when g(1) is called

```
program main ()
   var a, b : int;
   procedure f (m : int);
       var v: int:
      procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if(z > 4) f(z) else q(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f(4)
end main:
```



Answer! (cont.), when f(1) is called

```
program main ()
   var a, b : int;
 procedure f (m : int);
       var v: int:
       procedure g (y : int);
           var i, j : int;
           f (y);
       end g;
       procedure p (z : int);
           var I, m : int;
           procedure q (i : int);
              var k : int:
              g (i - 1);
              b := m + v;
           end q;
          if (z > 4) f (z) else a(z - 1)
       end p;
       if m > 1 then p(3)
   end f;
   f (4)
end main:
```



Answer! (cont.), when returned to g(1)

```
program main ()
   var a, b : int;
                                                                                      main()
   procedure f (m : int);
       var v: int:
                                                                                 a.b:
      procedure g (y : int);
                                                                                       f(4)
           var i, j : int;
           f (y);
                                                                                 m,v:
       end g;
                                                                                       p(3)
       procedure p (z : int);
           var I, m : int;
                                                       3:
                                                                                 z,1,m:
           procedure q (i : int);
                                                       4:
                                                                                       q(2)
               var k : int:
              g (i - 1);
                                                    Displays
                                                                                i,k:
              b := m + v;
           end q;
                                                                                       g(1)
          if(z > 4) f(z) else q(z - 1)
                                                                                 saved d[3]
       end p;
                                                                                 y,i,j:
       if m > 1 then p(3)
   end f;
   f (4)
                                                                                                      top
end main:
```

Answer! (cont.), when returned to q(2)

```
program main ()
   var a, b : int;
                                                                                   main()
   procedure f (m : int);
       var v: int:
                                                                              a.b:
       procedure g (y : int);
                                                                                    f(4)
           var i, j : int;
           f (y);
                                                                              m,v:
       end q;
                                                                                    p(3)
       procedure p (z : int);
           var I, m : int;
                                                     3:
                                                                              z,1,m:
       → procedure q (i : int);
                                                                                    q(2)
              var k : int:
              g (i - 1);
                                                   Displays
                                                                              i.k:
             b := m + v;
          end q;
          if (z > 4) f (z) else q(z - 1)
       end p;
                                                                                                  top
      if m > 1 then p(3)
   end f;
                                  address m: D[3] + #2
   f(4)
end main:
                                  address v: D[2] + #1
                                                                                             21
                                  address b: D[1] + #1
```