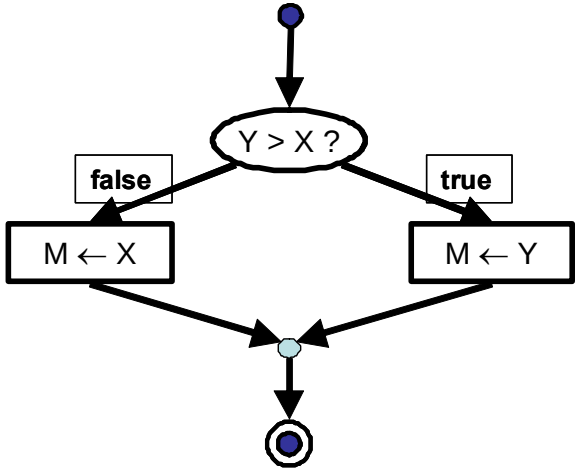
1. Section 5 Exercises

Program Memory

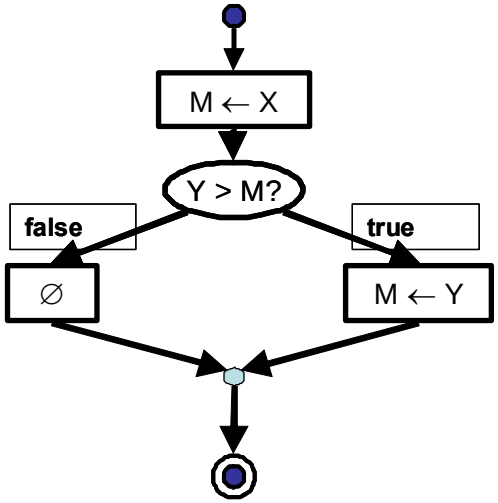
Exercise 5-1 - Back to the Larger of Two Numbers

Working memory

GIVENS: X, Y (two numbers)
RESULT: M (the larger of X and Y)
HEADER: M ← Max2(X, Y)
BODY:

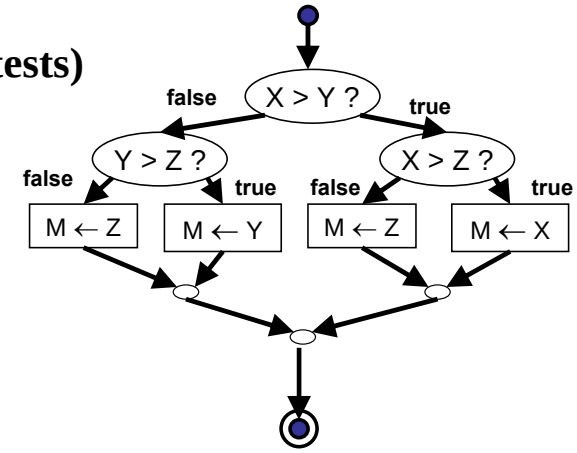


OR

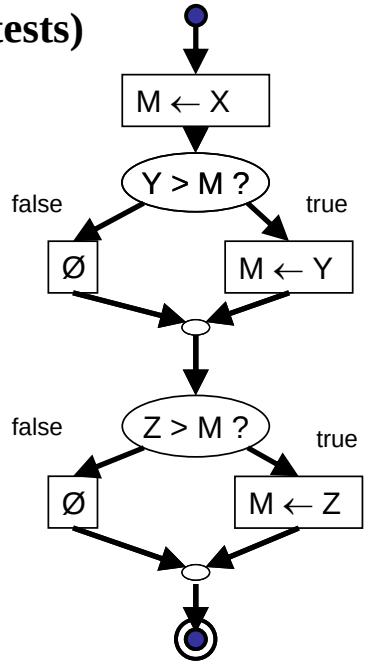


CPU

GIVENS: X, Y, Z (three numbers)
RESULT: M (the larger of X, Y and Z)
HEADER: M ← Max3(X, Y, Z)
BODY:
(Nested tests)



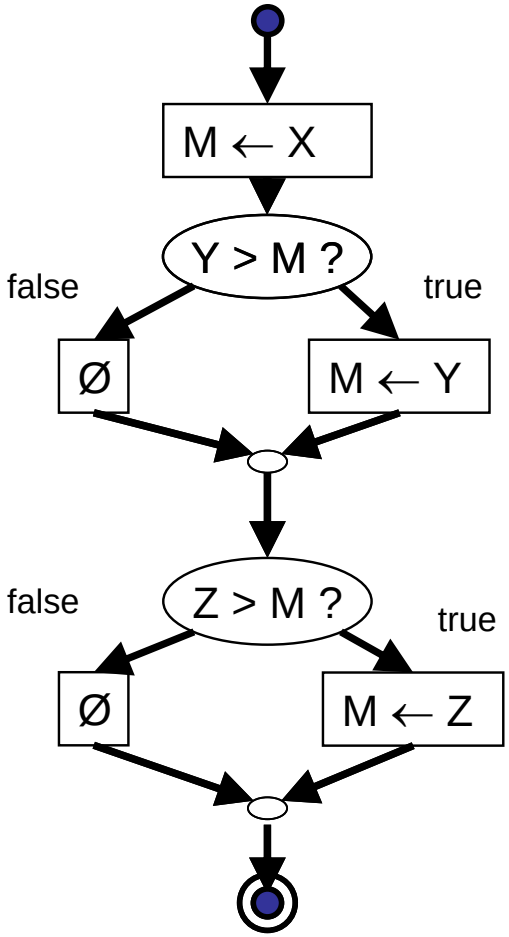
OR
(Sequence of tests)



CPU

GIVENS: X, Y, Z (three numbers)
RESULT: M (the larger of X, Y and Z)
HEADER: M ← Max3(X, Y, Z)
BODY:

Sequence of tests

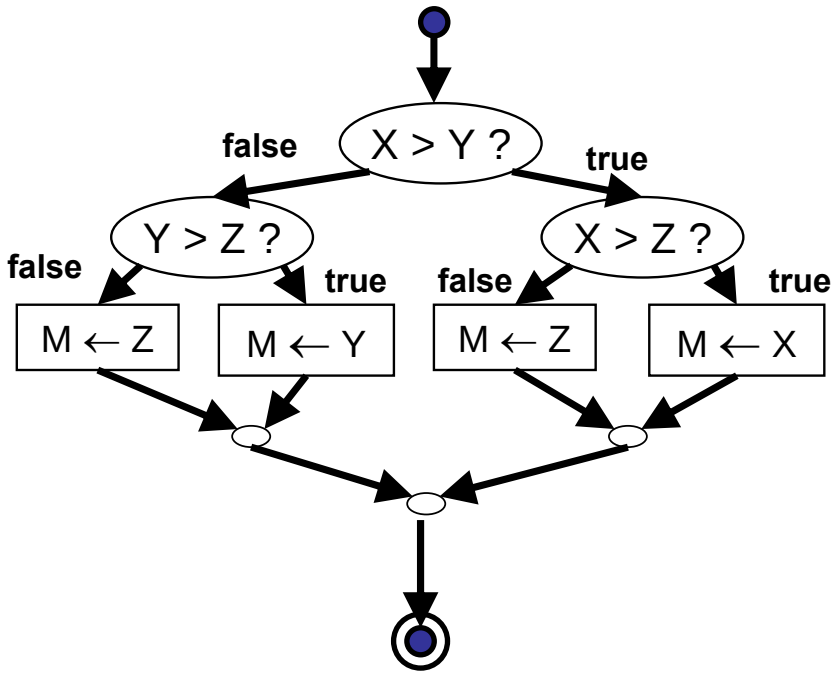



```
public double max3(double x,
          double y, double z)
{
    double m;
    m = x;
    if ( y > m )
    {
       m = y;
    }
    else
    {
       /* do nothing*/;
    }
    if ( z > m )
    {
       m = z;
    }
    else
    {
       /* do nothing*/;
    }
    return(m)
}
```

CPU

GIVENS: X, Y, Z (three numbers)
RESULT: M (the larger of X, Y and Z)
HEADER: M ← Max3(X, Y, Z)
BODY:

Nested tests



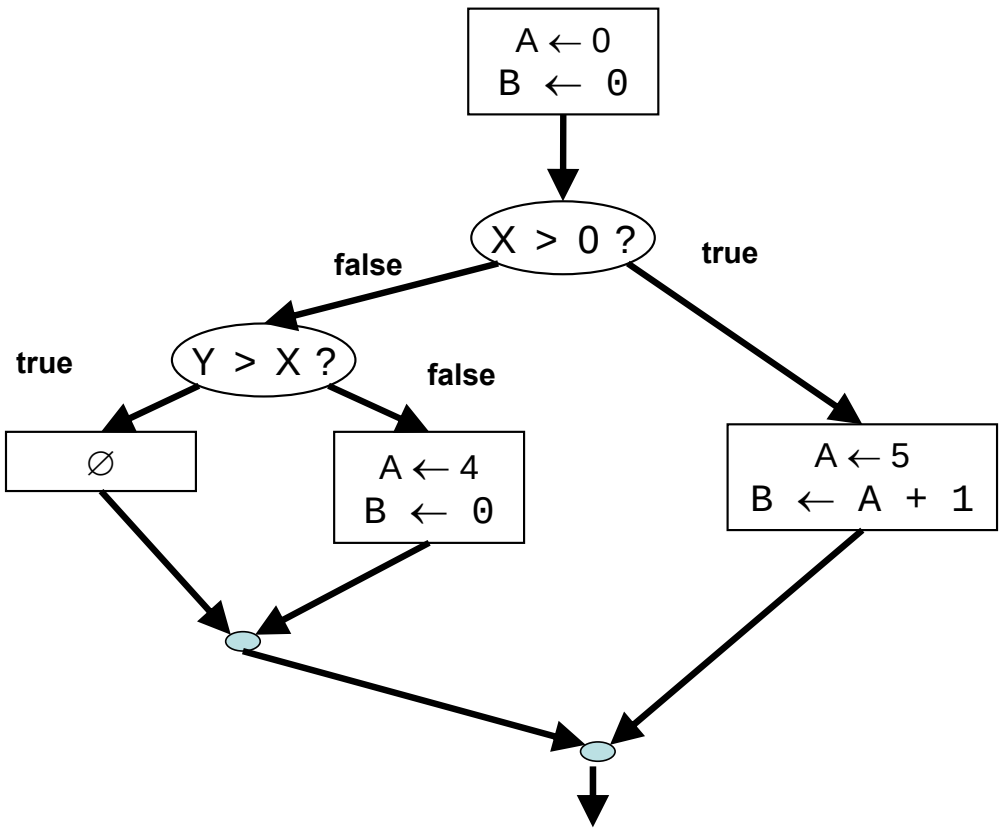

```
public double max3(double x,
        double y, double z)
{
    double m;
    if ( x > y )
    {
       if ( x > z )
       {
          m = x;
       }
       else
       {
          m = z;
       }
    }
    else
    {
       if ( y > z )
       {
          m = y;
       }
       else
       {
          m = z;
       }
    }
    return(m)
}
```

CPU

Program Memory

Exercise 5-5 – Translation of an Instruction Bloc

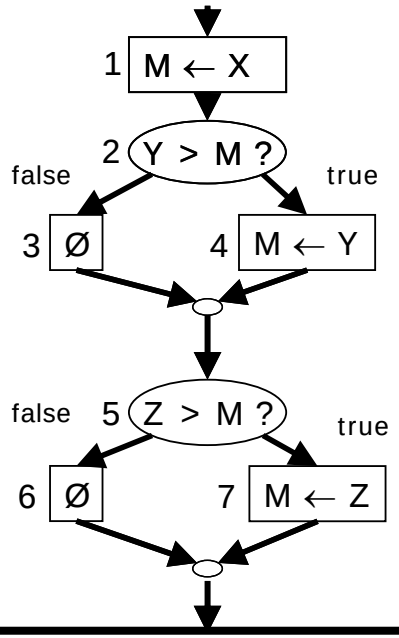
Working memory

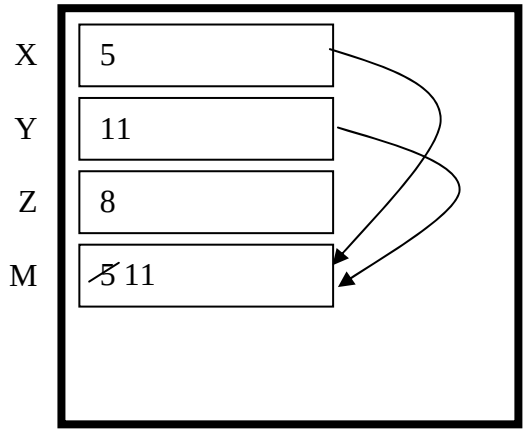



```
{
    a = 0;
    b = 0;
    if ( x > 0 )
    {
       a = 5;
       b = a + 1;
    }
    else
    {
       if ( y > x )
       {
          a = 4;
          b = 0;
       }
       else
       {
         /*do
    nothing*/;
       }
    }
}
```

CPU

GIVENS: X, Y, Z (three numbers)
RESULT: M (the larger of X, Y and Z)
HEADER: M ← Max3(X, Y, Z)
BODY:





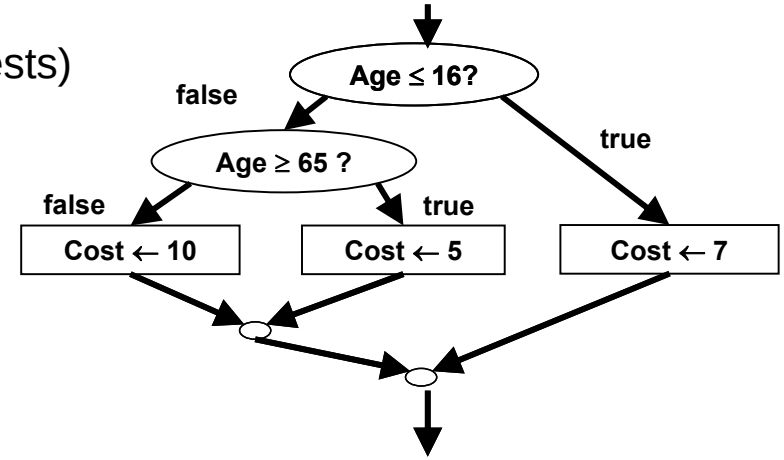
CPU

Table 1 – Trace for M ← Max3(5, 11, 8)

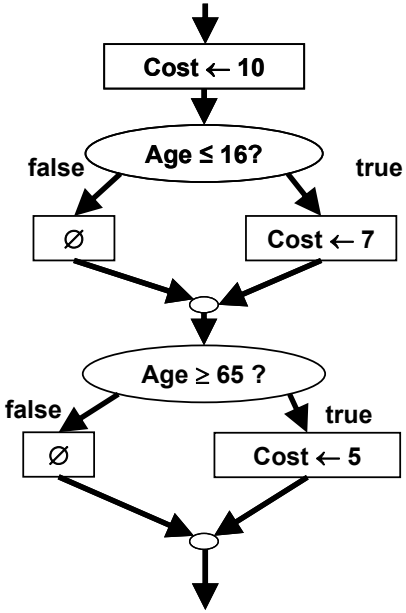
	X	Y	Z	M
Initial values	**5**	**11**	**8**	**?**
1. M ← X				**5**
2. Y > M: true				
3. M ← Y				**11**
4. Z > M: false				
5. ∅				

GIVENS: Age (persons age)
RESULT: Cost (ticket cost)
HEADER: Cost ← TicketCost(Age)
BODY:

(Version 1: Nested Tests)



(Version 2: Sequence of Tests)



CPU

Program Memory

Exercise 5-8 - Positive Value

Working memory

GIVEN: **X** **(a real number)**
RESULT: **Pos** **(a Boolean value, it is TRUE if and only if X is positive)**
HEADER: **Pos ← Positive (X)**
BODY
Pos ← (X > 0)

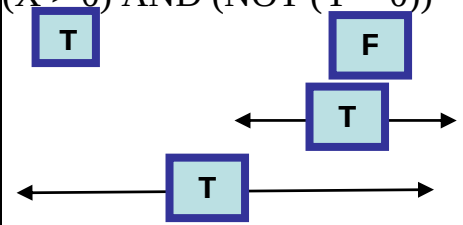
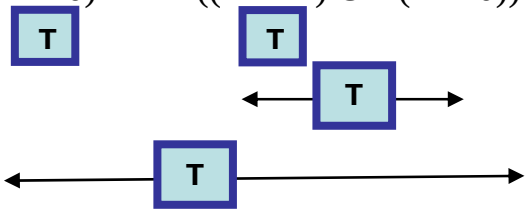
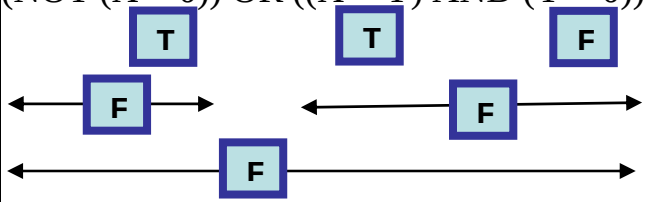
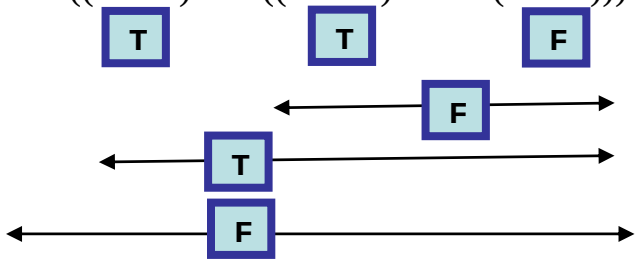
CPU

Exercise 5-9 Compound Boolean Expressions

Age > 16 AND Age < 65

Exercise 5-10 - More Compound Boolean Expressions

Suppose X = 5 and Y = 10.

Expression	Value
(X > 0) AND (NOT (Y = 0)) 	TRUE
(X > 0) AND ((X < Y) OR (Y = 0)) 	TRUE
(NOT (X > 0)) OR ((X < Y) AND (Y = 0)) 	FALSE
NOT ((X > 0) OR ((X < Y) AND (Y = 0))) 	FALSE