1. Section 5 Exercises

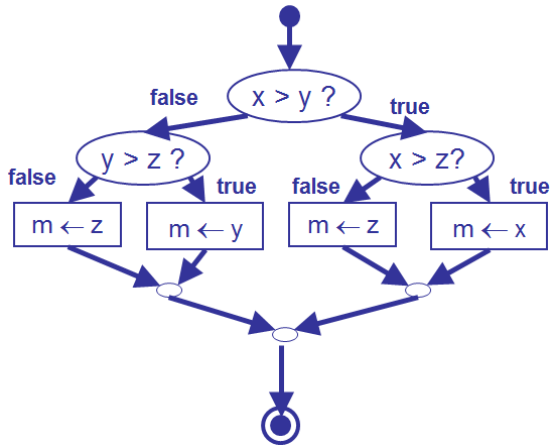**GIVENS:**   x, y (two numbers)

RESULT: m   (the larger of x and y)

HEADER: m ← max2(x, y)
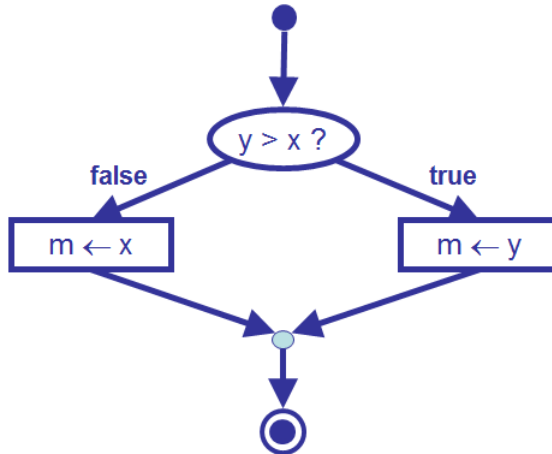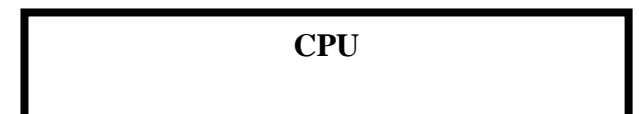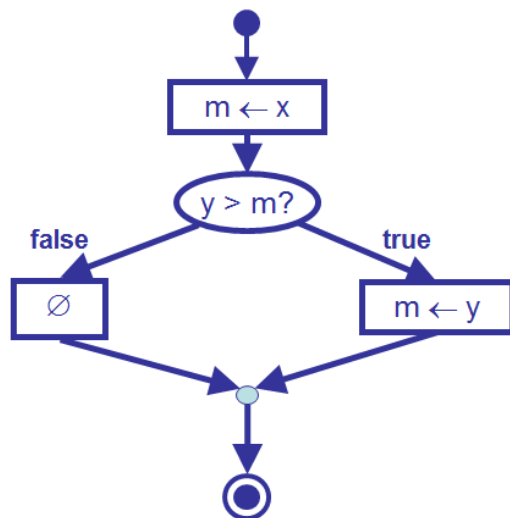
BODY:

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)

x > y ?

false → y > z ?

y > z ? false → m ← z

y > z ? true → m ← y

x > y ? true → x > z?

x > z? false → m ← z

x > z? true → m ← x

OR (Sequence of tests)

m ← x

y > m ?

false → Ø
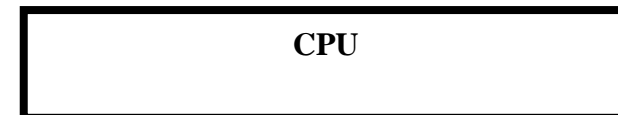
true → m ← y

z > m ?

false → Ø

true → m ← z

CPU

2
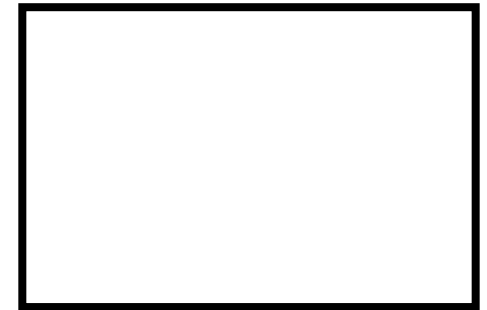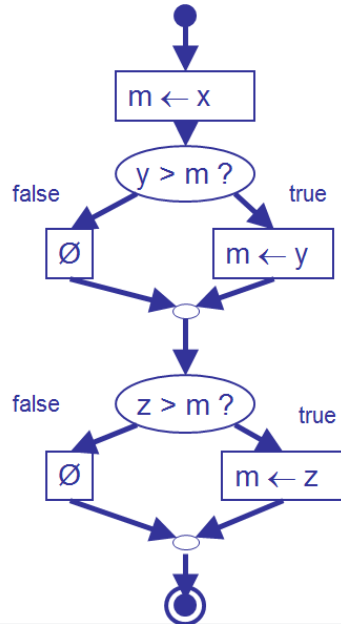
GIVENS: x, y, z (three numbers)
RESULT: m (the larger of x, y and z)
HEADER: m ← max3(x, y, z)
BODY:

Sequence of tests
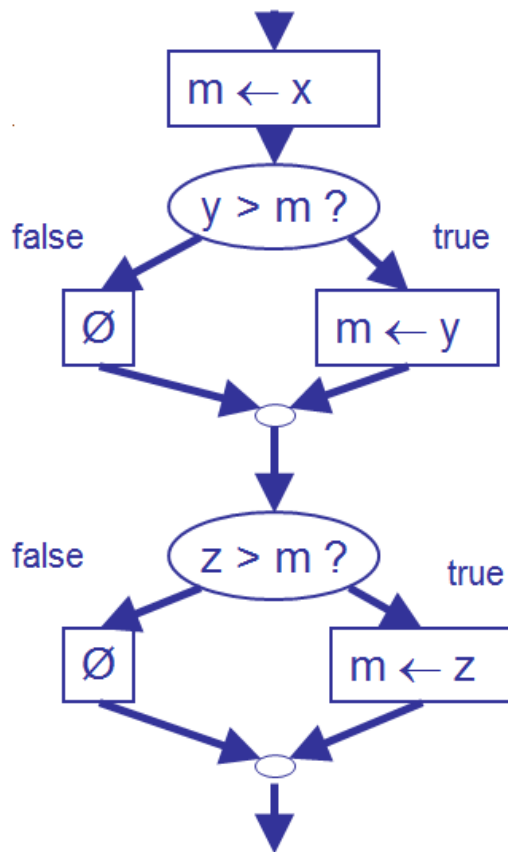

```java
public double max3(double x,
          double y, double z)
{
    double m;
    m = x;
    if ( y > m )
    {
        m = y;
    }
    else
    {
        /* do nothing*/;
    }
    if ( z > m )
    {
        m = z;
    }
    else
    {
        /* do nothing*/;
    }
    return m;
}
```
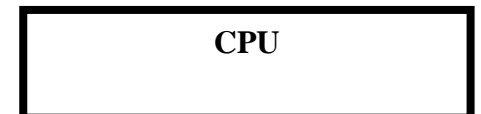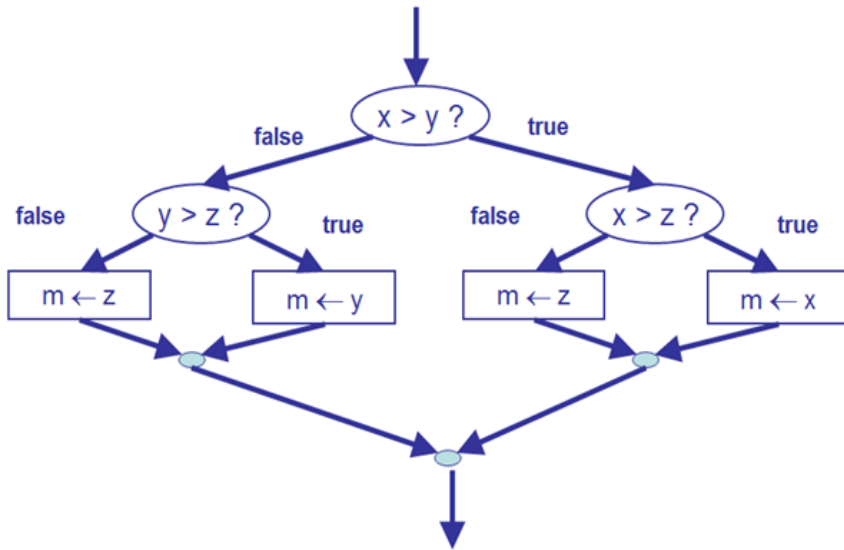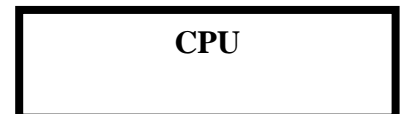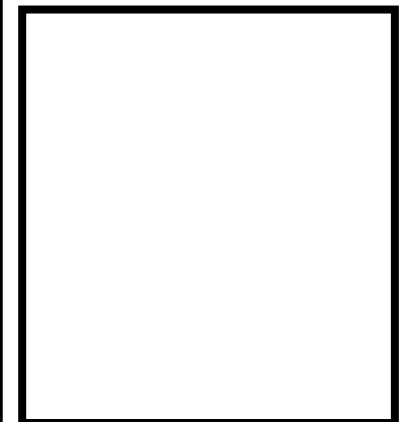
CPU

3

GIVENS: x, y, z (three numbers)
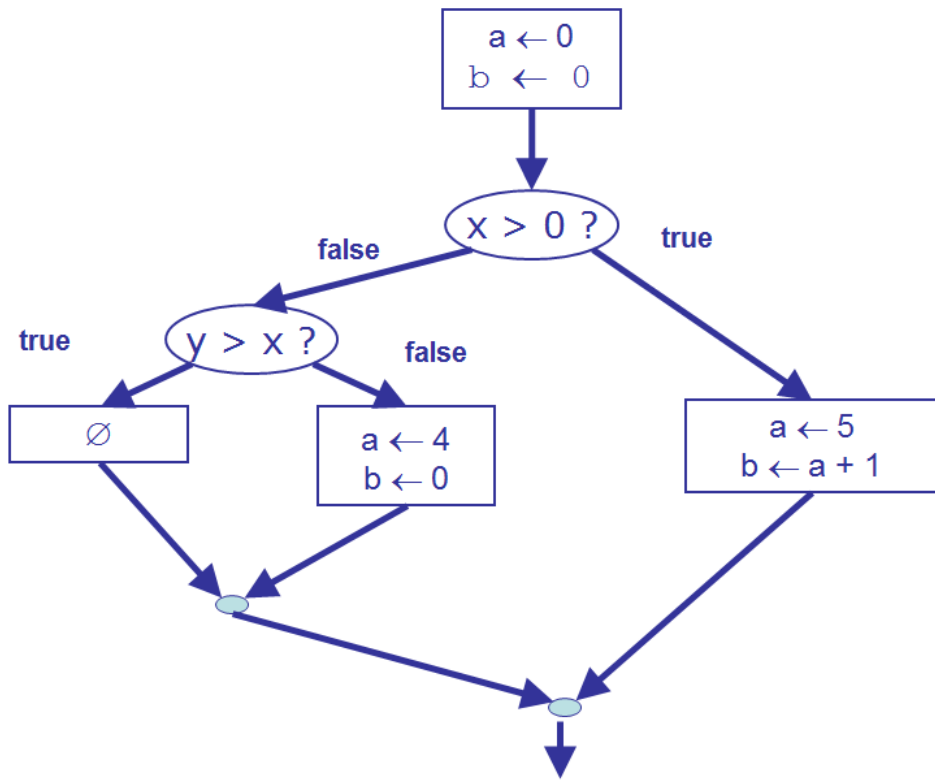**RESULT: m (the larger of x, y and z)**
HEADER: m ← max3(x, y, z)
BODY:

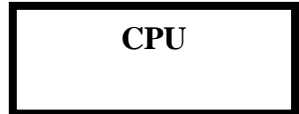**Nested tests**


```
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;
}
```
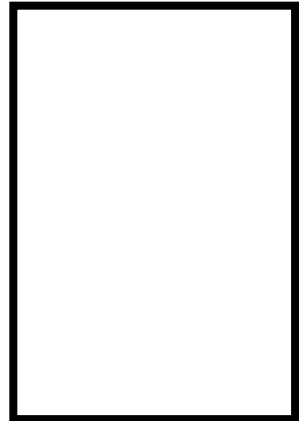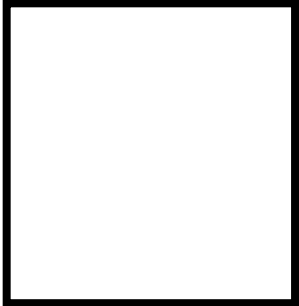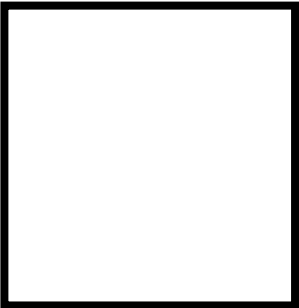
CPU

4

```
a ← 0
b ← 0
```

```
x > 0 ?
```

false   true

```
y > x ?
```

true   false

```
∅
```

```
a ← 4
b ← 0
```

```
a ← 5
b ← a + 1
```

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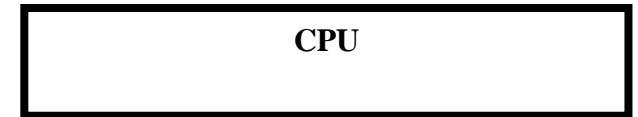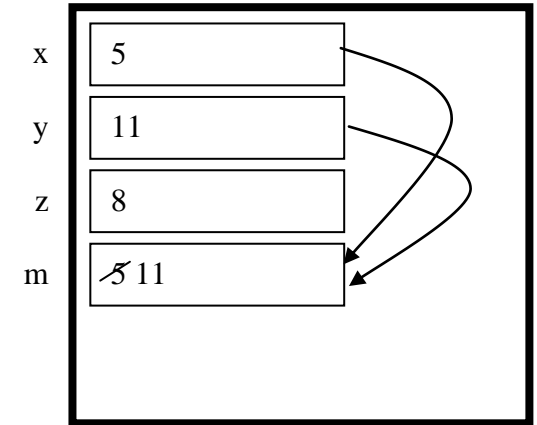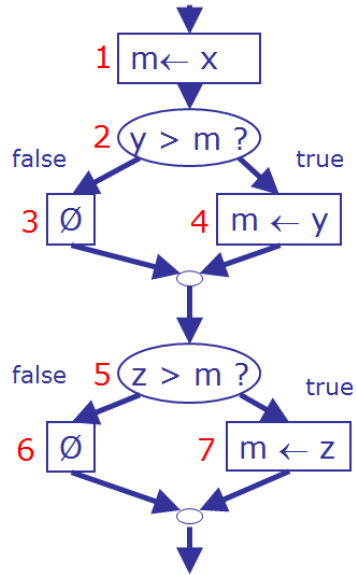
CPU

GIVENS: x, y, z   (three numbers)
RESULT: m   (the larger of x, y and z)
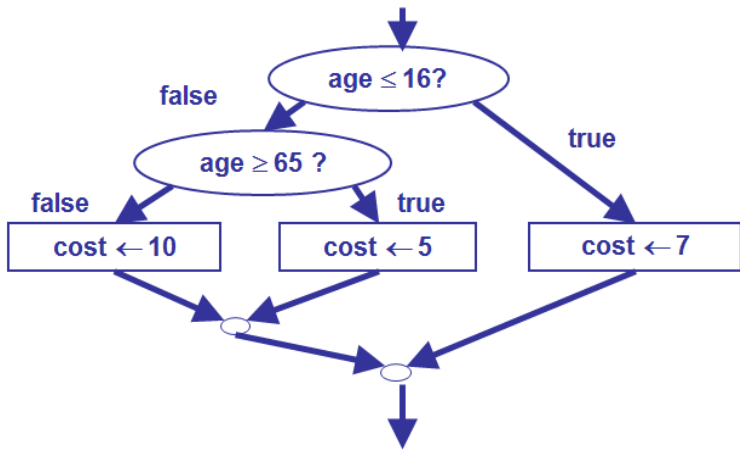**HEADER:**   m ← max3(x, y, z)
BODY:

1. m ← x
2. y > m ?
    false   true
3. Ø   4. m ← y

5. z > m ?
    false   true
6. Ø   7. m ← z

Working memory:
x : 5
y : 11
z : 8
m : 5̶ 11

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. Ø				

6

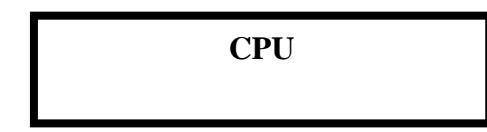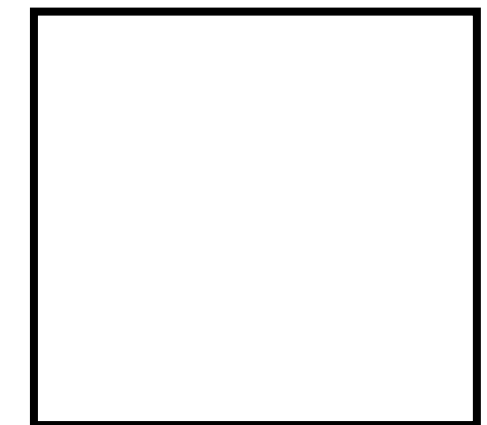GIVENS: age (persons age)
RESULT: cost   (ticket cost)
HEADER: cost $\leftarrow$ ticketCost(age)
BODY:

(Version 1: Nested Tests)
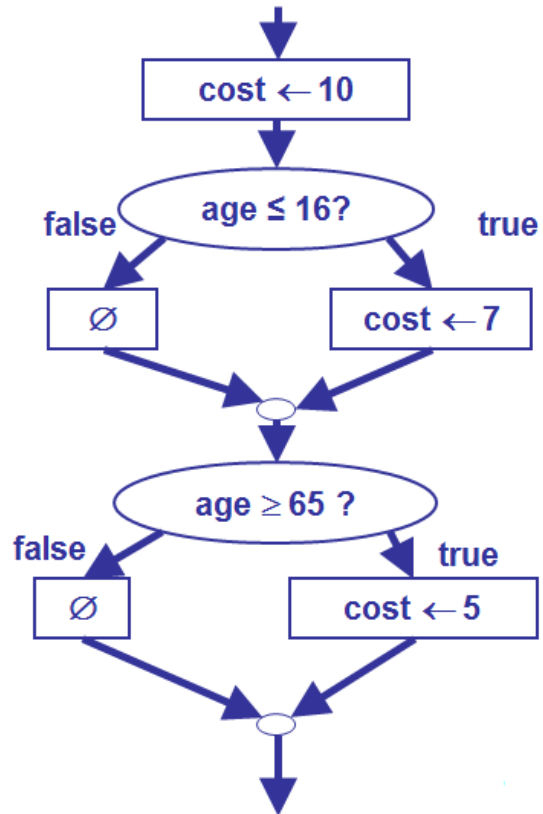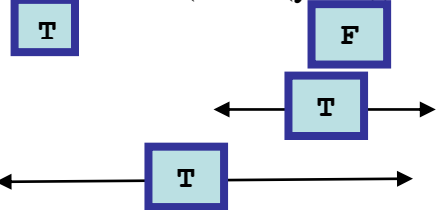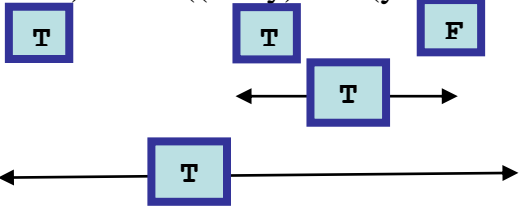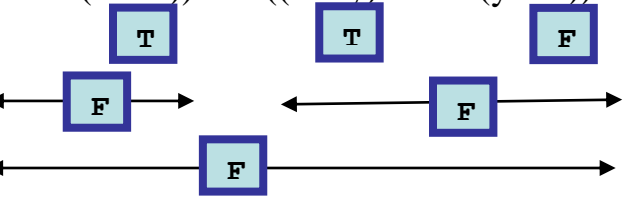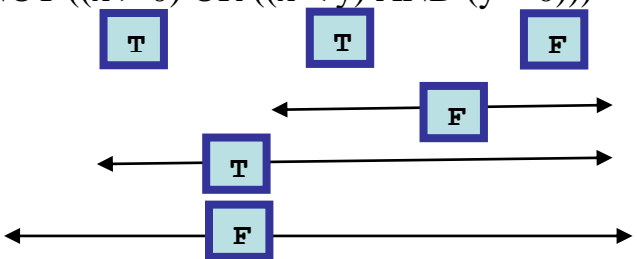
(Version 2: sequence of tests)

CPU

GIVEN: **x**   **(a real number)**
RESULT: ρ**os**   **(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)) T F T T	TRUE
(x > 0) AND ((x < y) OR (y = 0)) T T F T T	TRUE
(NOT (x > 0)) OR ((x < y) AND (y = 0)) T T F F F F	FALSE
NOT ((x > 0) OR ((x < y) AND (y = 0))) T T F F T F	FALSE