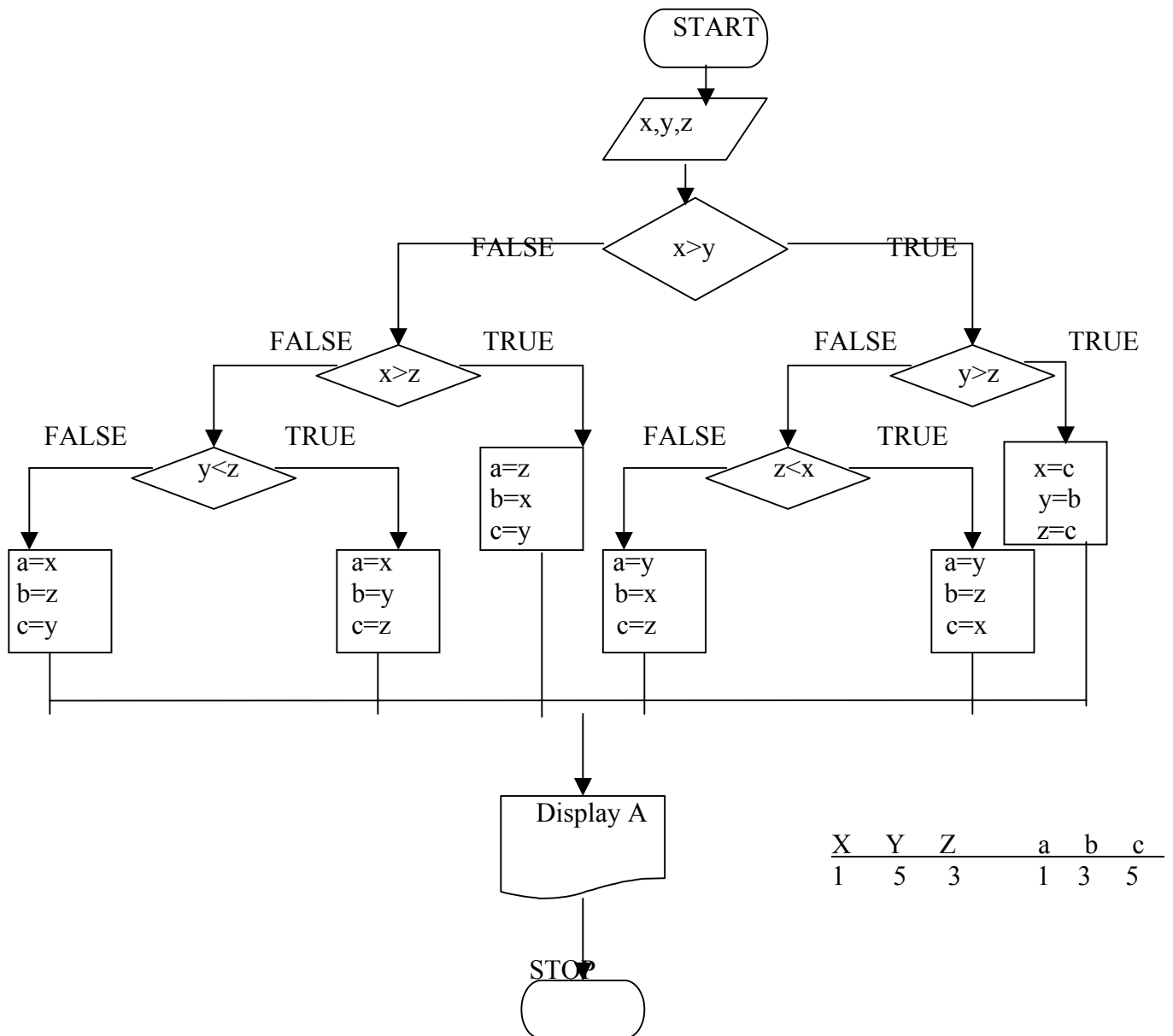
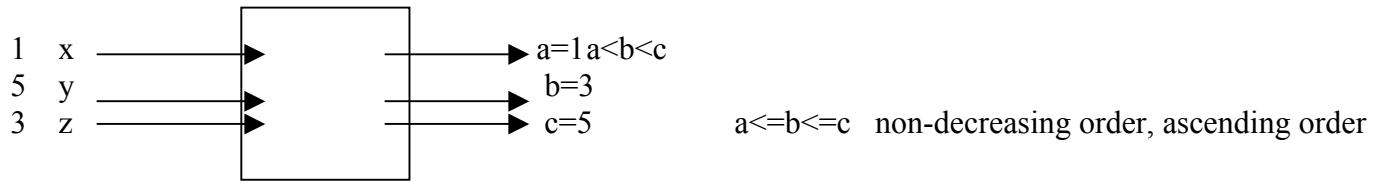
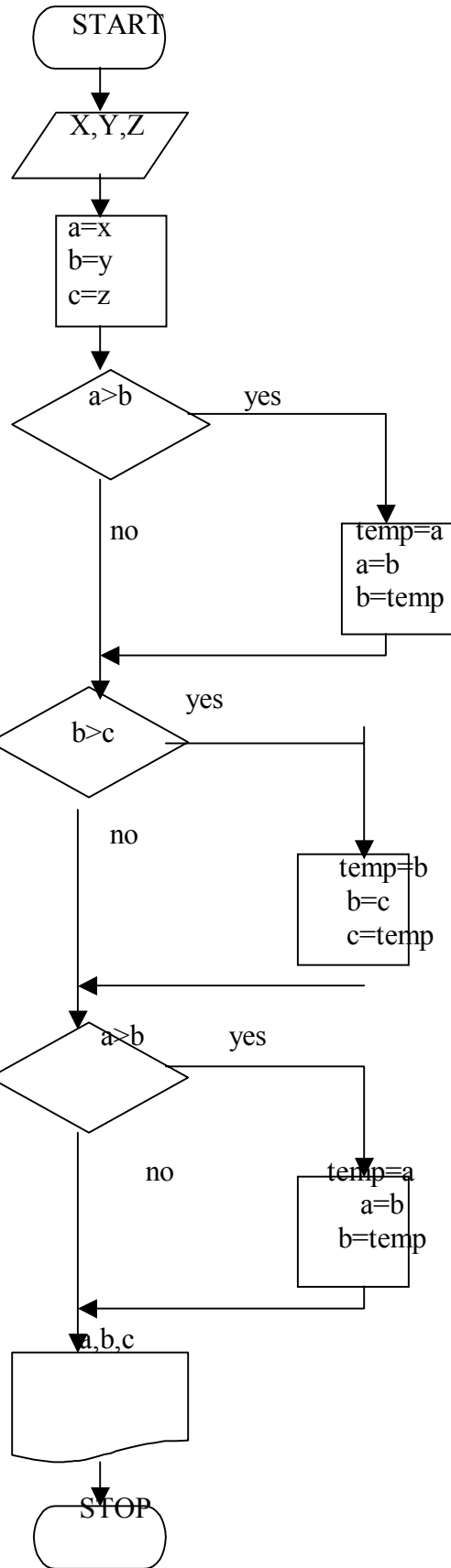


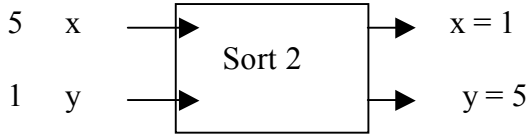
Problem: Sort three numbers in ascending (increasing) order





<u>X</u>	<u>Y</u>	<u>Z</u>
5	3	1

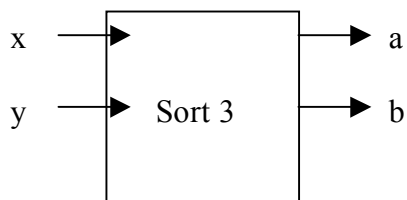
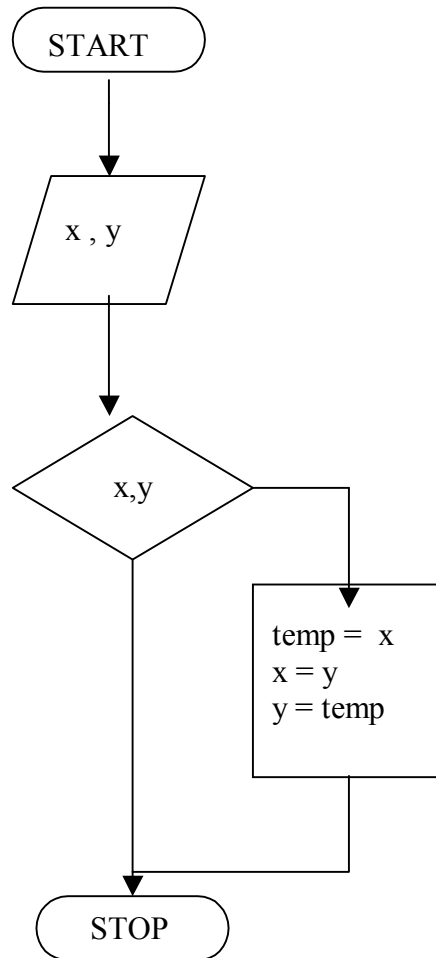
<u>a</u>	<u>b</u>	<u>c</u>	<u>temp</u>
5	3	1	5
3	5	5	3
1	1		
	3		



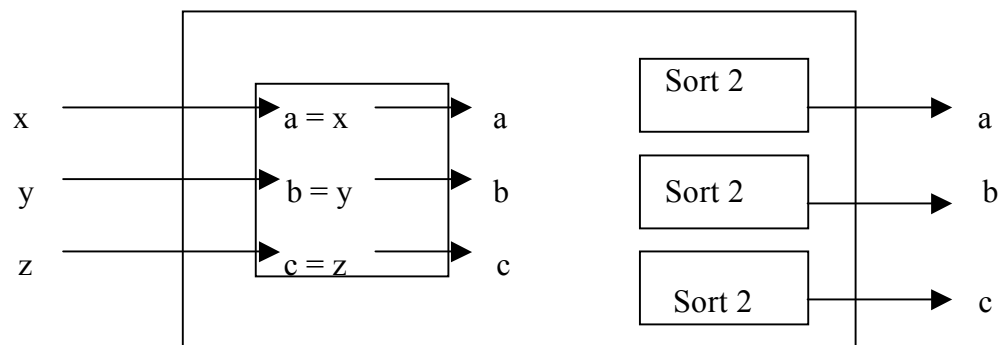
START
Read x, y
If $x > y$ then
 $\text{temp} = x$
 $x = y$
 $y = \text{temp}$
endif
STOP

This is the idea which we
are following through this chart.

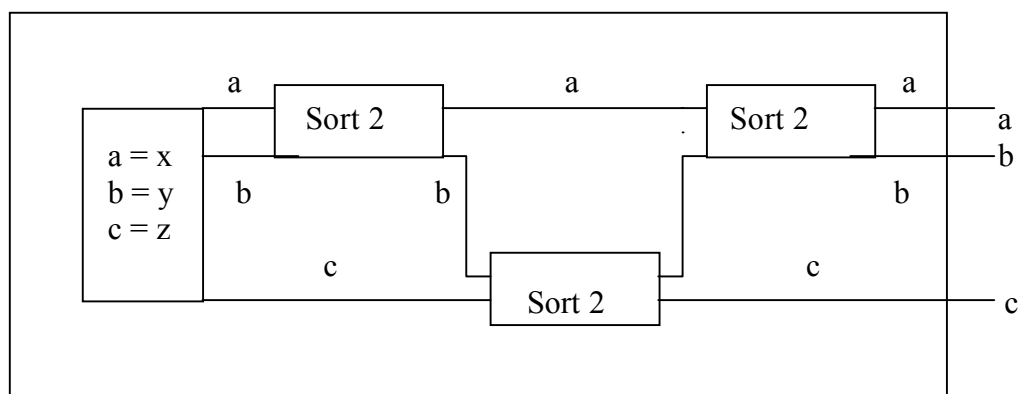
<u>x</u>	<u>y</u>	<u>temp</u>
5	1	5
1	5	

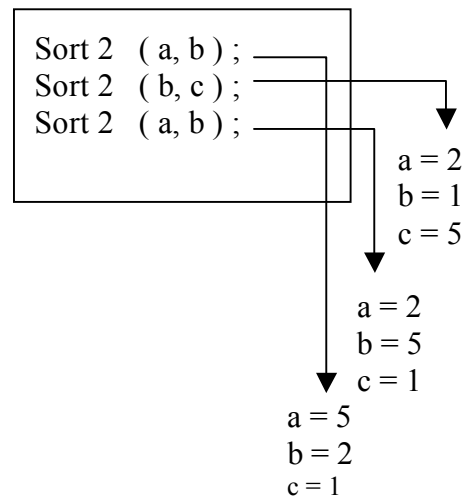
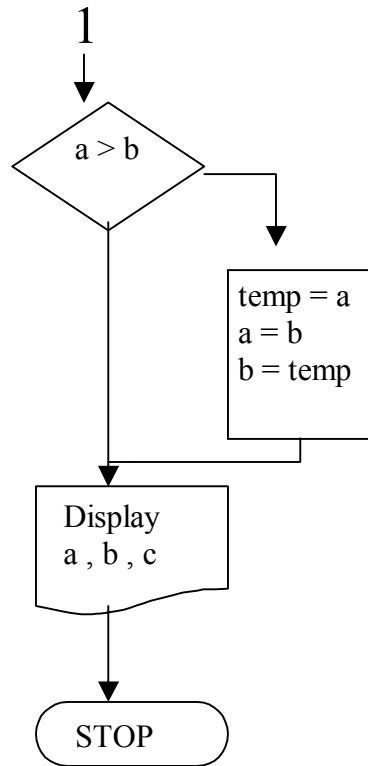
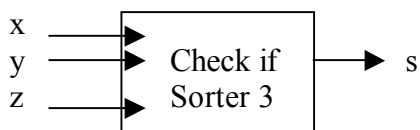
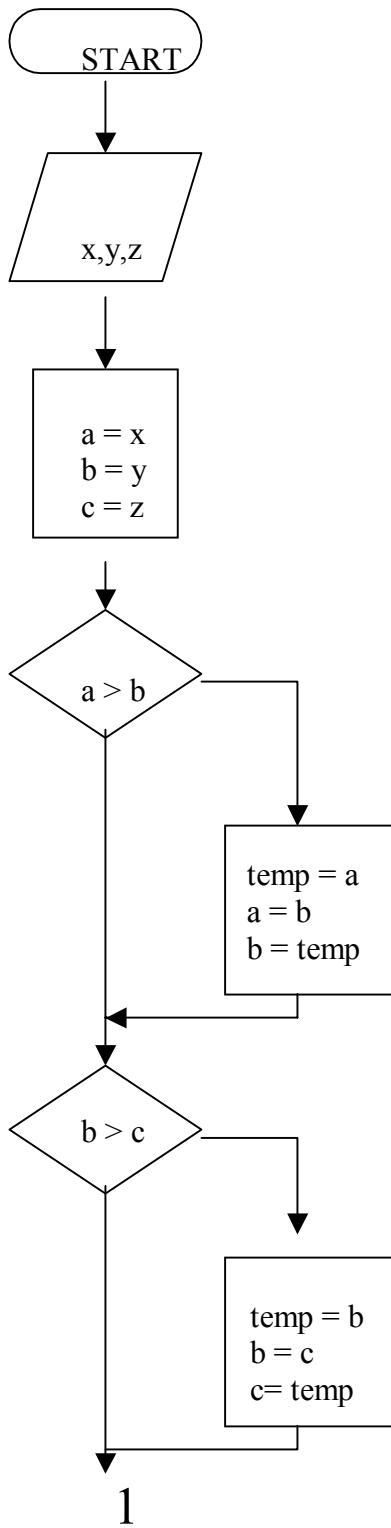


$a \leq b \leq c$



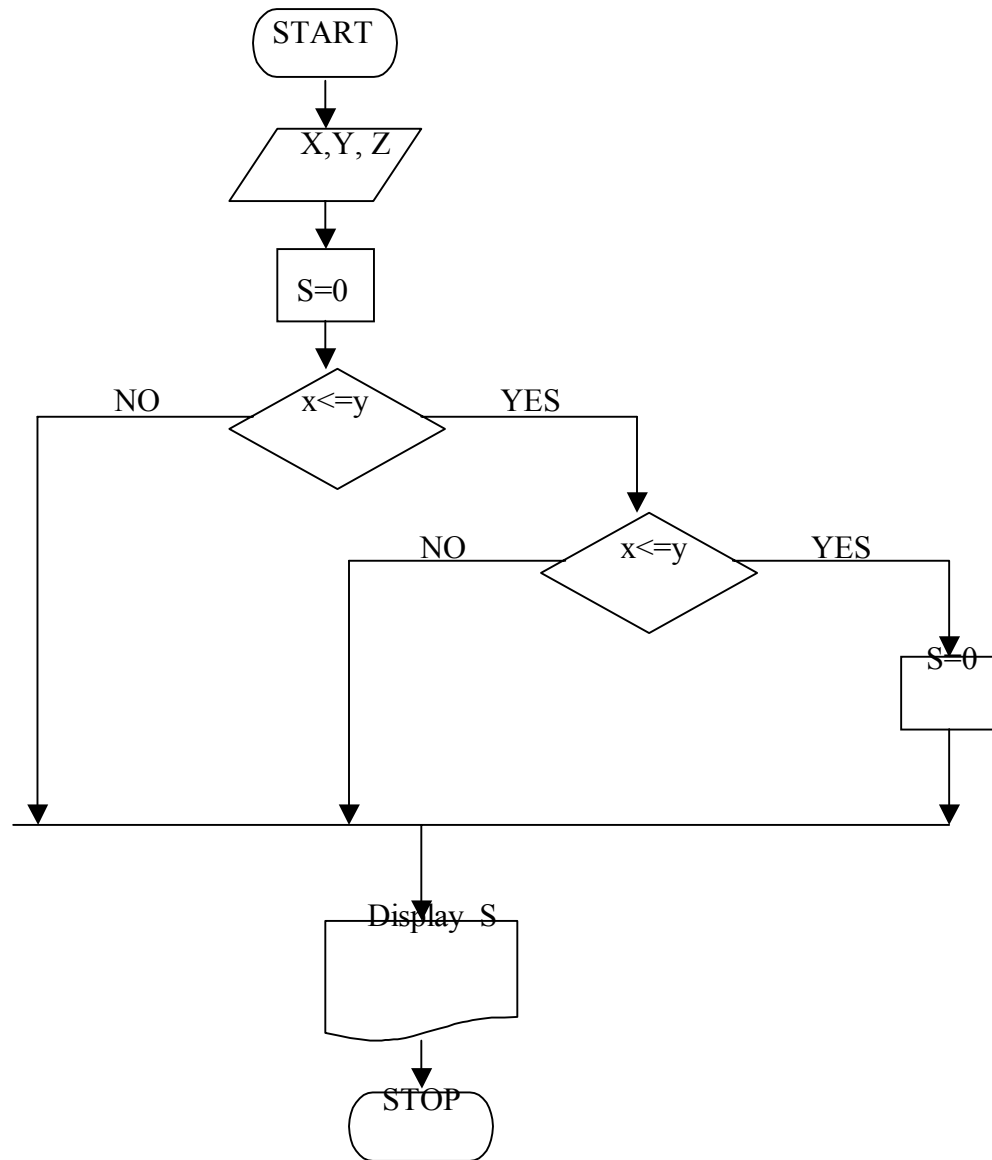
$a \leq b \leq c$





Problem: Test if three input numbers are in ascending

$$s = \begin{cases} 1 & \text{if } x \leq y \leq z \\ 0 & \text{otherwise} \end{cases}$$



START

Read x,y,z

S=0 assume unsorted

If x<=y then

If y<=z then

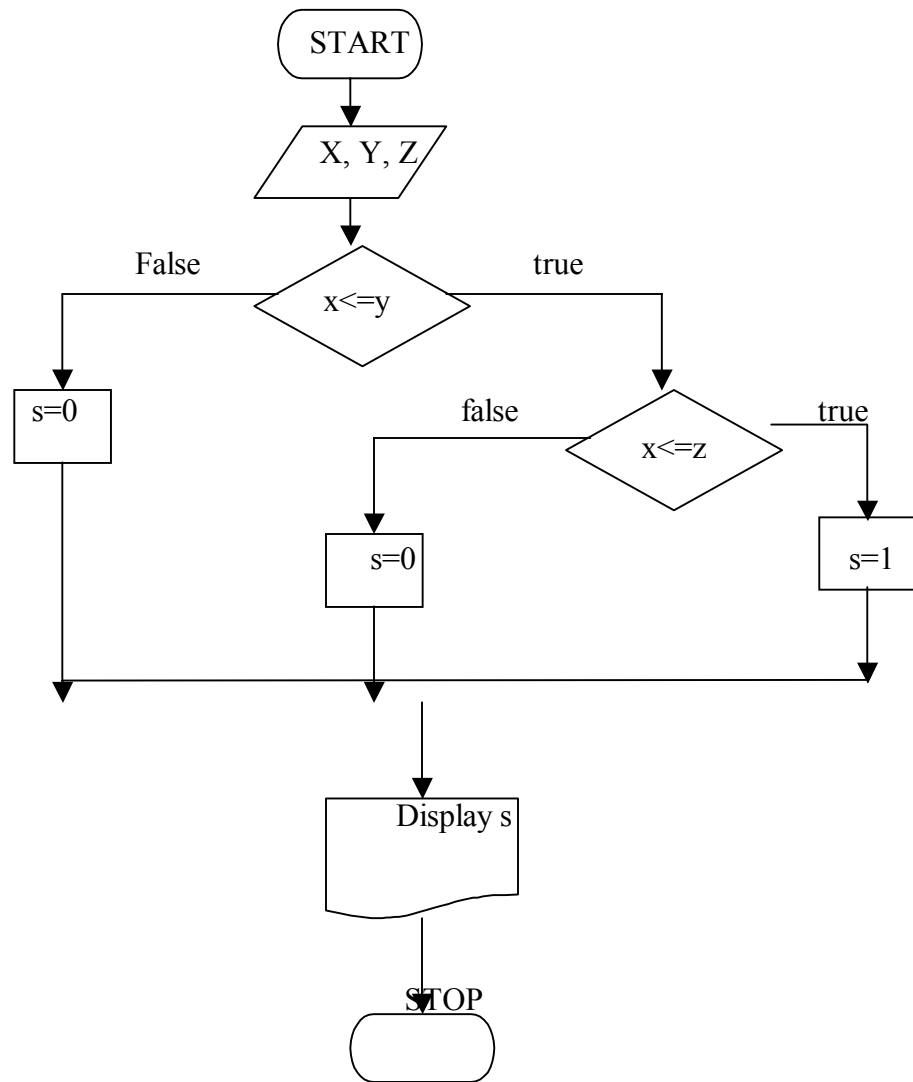
S=1

End if

End if

Display S

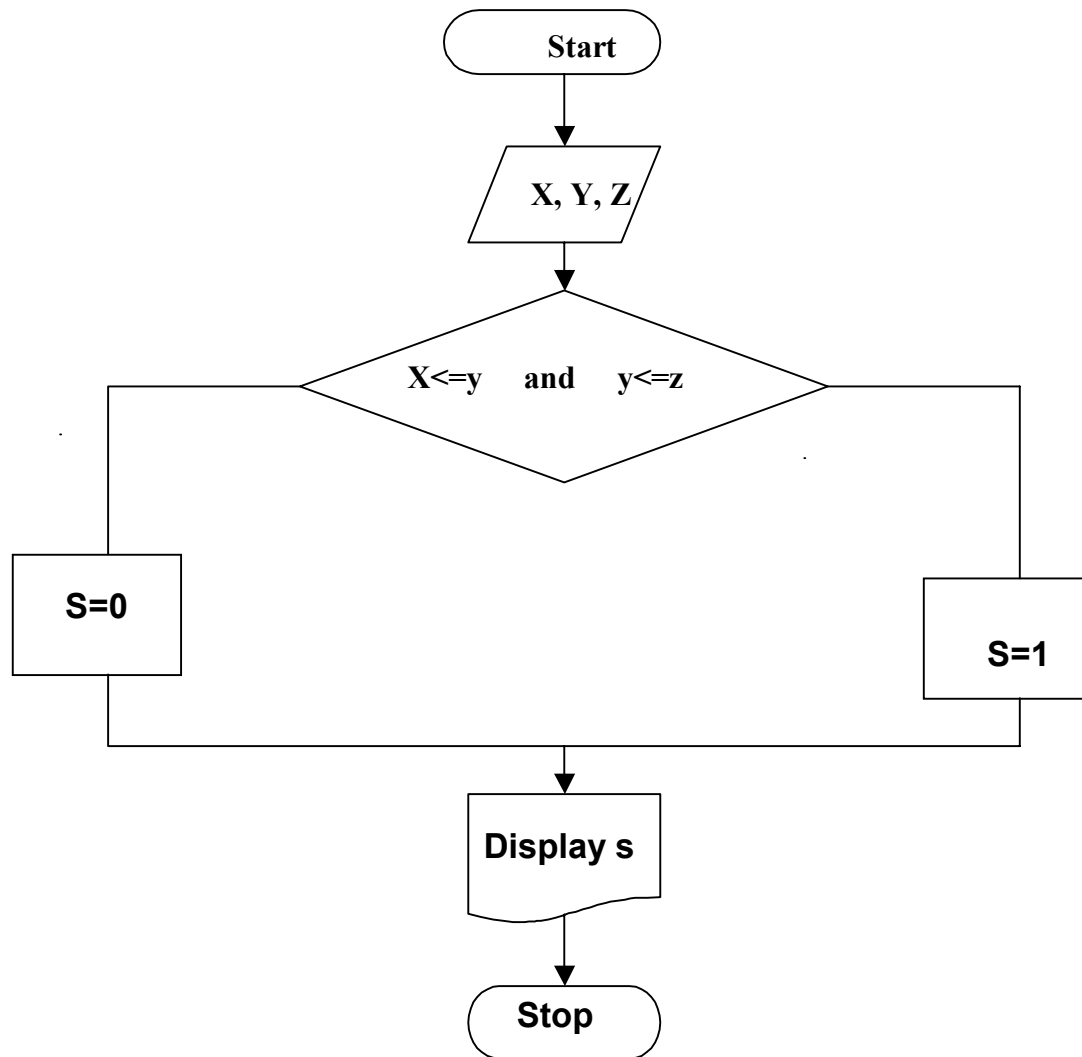
STOP



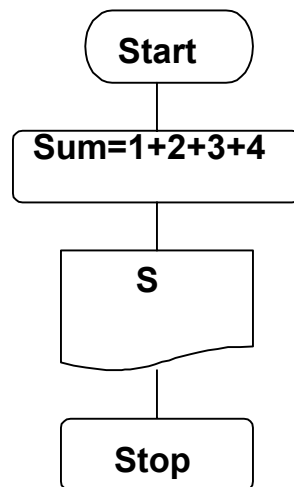
```

START
Read  x,y,z
If   x<=y then
    If   y<=z then
        s=1
    else
        s=0
    end if
else
    s=0
end if
Display
STOP
  
```

ANOTHER SOLUTION

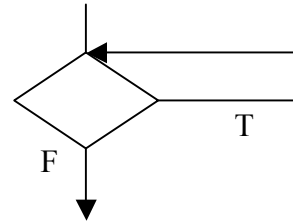
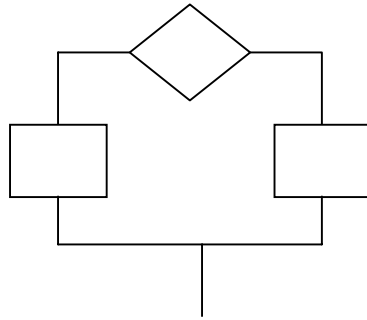
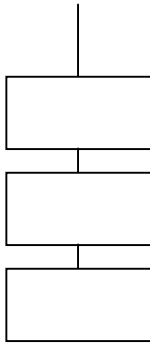


Find summation of numbers from 1 to 4

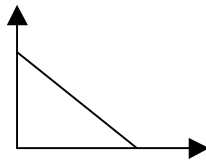


PROGRAMMING STRUCTURE

1. Sequence
2. Selection
3. Iteration



Memory



Time

Efficiency

1. Time efficiency
2. Memory efficiency

Consider the following due to Gauss

$$S = 1 + 2 + 3 + \dots + n$$

$$+ S = n + (n-1) + (n-2) + \dots + 1$$

$$2S = (n+1) + (n+1) + \dots + (n+1)$$

$$S = n \times (n+1) / 2$$

Pseudo Code

START

READ n

IF n > 0 then

 Sum = n.(n+1) / 2

 DISPLAY SUM

Else

 DISPLAY ERROR n <= 0

END IF

STOP

Or

PROBLEMS

1. Find the sum of even numbers from 0 to n

```
START
READ n
IF n> 0 then
    Sum=0
    i=0
    L:if i<=n then
        Sum=sum+i
        i=i+2
        Go to L
    END IF
    DISPLAY SUM
Else // n<=0
DISPLAY error n<=0
END IF
STOP
```

2. Find the maximum of three numbers<x,y,z>

```
START
READ x,y,z
IF x>y then
    if z>x
        A=z
    Else
        A=x
ELSE
    if z>y then
        A=z
    Else
        A=y
END IF
STOP
```

3. Find the avarage of n numbers

```
START
READ n
IF n>0 then
    Sum= n.(n+1)/2
    A=sum/n
ELSE // n<=0
DISPLAY error n<=0
END IF
STOP
```

5. FIND THE SUM OF ODD NUMBERS FROM 1 TO N

Start

Read n

If n > 0 then

Sum=0

I= 0

L: if i<=n then

Sum= sum+i

I=I+1

Go to l

End if

Display sum

Else // n<=0

Display error n<=0

End if

Stop