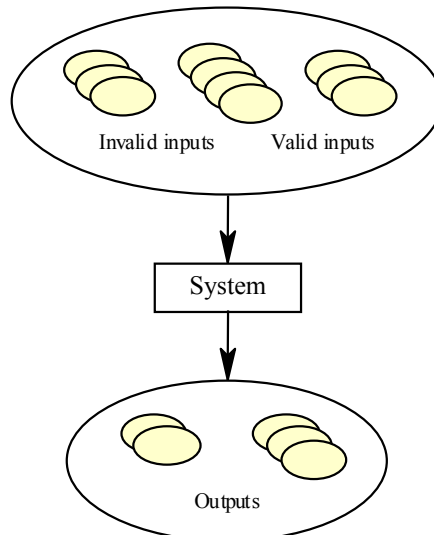


## Equivalence partitioning



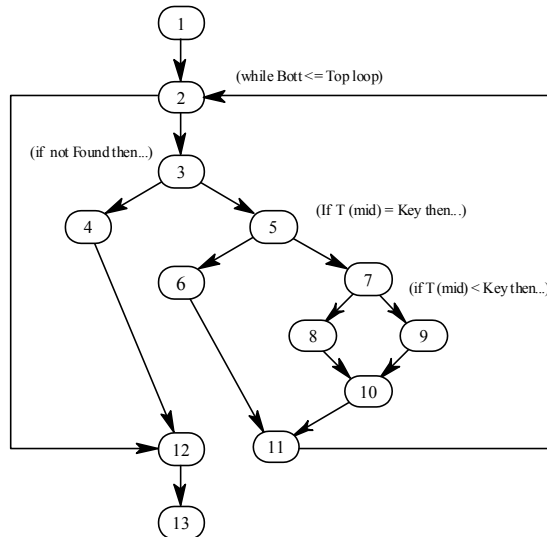
©Ian Sommerville 1995 Software Engineering, 5th edition. Chapter 23 Slide 12

## Binary search (C++)

```
void Binary_search (elem key, elem* T, int size,
                   boolean &found, int &L)
{
    int bott, top, mid ;
    bott = 0 ; top = size -1 ;
    L = ( top + bott ) / 2 ;
    if (T[L] == key)
        found = true ;
    else
        found = false ;
    while (bott <=top && !found)
    {
        mid = (top + bott) / 2 ;
        if ( T [mid] == key )
        {
            found = true;
            L = mid ;
        }
        else if (T [mid] < key )
            bott = mid + 1 ;
        else
            top = mid-1 ;
    } // while
} //binary_search
```

©Ian Sommerville 1995 Software Engineering, 5th edition. Chapter 23 Slide 23

## Binary search flow graph



## Control and data-driven programs

```
case A is
  when "One" => i := 1 ;
  when "Two" => i := 2 ;
  when "Three" => i := 3 ;
  when "Four" => i := 4 ;
  when "Five" => i := 5 ;
end case ;
```

```
Strings: array (1..4) of STRING :=
  ("One", "Two", "Three", "Four", "Five");
i := 1 ;
loop
  exit when Strings (i) = A ;
  i := i + 1 ;
end loop ;
```