

## DIT636 / DAT560 - Structural Testing Activity

1. Draw a control-flow graph for the following method:  
(“where” is an array, “what” is the element you are searching for)

```
1. public int search(String[] where, String what){
2.     int index = 0;
3.     if ((where.length == 1) && (where[0] == what)){
4.         return 0;
5.     } else if (where.length == 0){
6.         return -1;
7.     } else if (where.length > 1){
8.         while(index < where.length){
9.             if (where[index] == what){
10.                return index;
11.            } else
12.                index++;
13.        }
14.    }
15. }
16. return -1;
17. }
```

2. Select test input that provides statement, branch, and basic condition coverage over the code.

You do not need to write full unit tests. Just state the input, and explain which lines and/or branches are covered by the code. For example:

[ ] (empty array), "Bob"

Executes lines 1, 2, 3, (Branch 3-F), 5, (Branch 5-T), 6

## DIT636 / DAT560 - Loop Testing Activity

1. Draw the control-flow graph for the following code:

```
1. public boolean binary_search (Object key, Object[] T){
2.     int mid;
3.     int bott = 0;
4.     int top = T.length - 1;
5.     Boolean found = false;
6.
7.     if(T[0] == key){
8.         found = true;
9.     }else{
10.        found = false;
11.    }
12.
13.    while (bott <=top && !found){
14.        mid = round((top + bott) / 2);
15.        if(T[mid] == key){
16.            found = true;
17.        } else{
18.            if (T [mid] < key ){
19.                bott = mid + 1;
20.            }else{
21.                top = mid-1;
22.            }
23.        }
24.    }
25.    return found;
26.}
```

**2. Identify test input that achieves loop boundary coverage**

**(That exercises the loops:**

- Zero times**
- One time**
- Two or more times)**

**Again, you do not need to create full unit tests. Simply select input and explain how it exercises the loops. For example:**

**1, [1]**

**Executes the loop 0 times.**