

### Homework #3

## CENG 437 – Software Quality Management

In this homework you are expected to perform Data Flow Testing. You are given two binary search methods (`binarySearch1` and `binarySearch2`) written in Java language. They both take the same input parameters. The input array “array” is assumed to be sorted and the input “x” is the search element. Both method returns the index of the searched element of “x”.

```
public static int binarySearch1 (int x, int [] array)
{
    int i = 0, j = array.length - 1;

    while (j >= i)
    {
        int m = (i + j) / 2;

        if (array[m] > x)
            j = m - 1;
        else if (array[m] < x)
            i = m + 1;
        else
            return m;
    }

    return -1;
}

public static int binarySearch2 (int x, int [] array)
{
    int i = 0, j = array.length - 1;

    while (j >= i)
    {
        int m = (i + j) / 2;

        if (array[m] > x) {
            j = m - 1;
            m = m - 1;
        }
        else if (array[m] < x)
            i = m + 1;
        else
            return m;
    }

    return -1;
}
```

## Questions

Please solve and answer the following questions given below:

1. Draw a data flow graph for the `binarySearch1()` function given in
2. Assuming that the input `array[]` has at least one element in it, find an infeasible path in the data flow graph for the `binarySearch1()` function.
3. Identify a data flow anomaly in the code `binarySearch2()`.
4. By referring to the data flow graph obtained from `binarySearch1()`, find a set of complete paths satisfying the all-defs selection criterion with respect to variable "m".
5. By referring to the data flow graph obtained from `binarySearch1()`, find a set of complete paths satisfying the all-defs selection criterion with respect to variable "j".
6. By referring to the data flow graph obtained from `binarySearch2()`, find a set of complete paths satisfying the all-defs selection criterion with respect to variable "m".
7. Program anomaly has been defined by considering three operations, namely, **define** (*d*), **reference** (*r*), and **undefine** (*u*). The three sequences of operations identified to be program anomaly are *dd*, *du*, and *ur*. Briefly explain why each three-operation sequences are not considered to be program anomaly. In addition, give a small Java code snippet example to each program anomaly.

### Submission Rules:

- **Due Date:** 30.03.2019, 23:55
- If any cheating is detected in your homework, will be graded as 0.
- Please submit your homework through CMS by exporting your Java Project.
- Please export your Java Project and homework document as the given format with your student ID: **StdID\_CENG437\_HW03.zip**.