Prof. Dr.-Ing. Peter Liggesmeyer
Dipl.-Inf. Max Steiner

TU Kaiserslautern
Dept. of Computer Science
AG Software Engineering: Dependability

Software Quality Assurance (WS14/15)

Problem Set 3

Due: in exercise, 10.12.2014

Problem 1: Data Flow Oriented Test

Given is the function `sum`:

```
public static int sum(int n) {
 int sum = 0;
 int i;
 for (i = 1; i <= n; i++) {
   sum = sum + i;
 }
 return sum;
}
```

a) Please create a control flow diagram with data flow annotation for the function `sum`.
b) Write down all def-use pairs in a table as in the example below. Indicate p-uses and c-uses.
c) Please determine the minimal necessary test path for fulfilling the ***all defs*** criterion of the `sum` function. Please denote the required test path and mark this path in the control flow diagram.
d) Please determine the minimal necessary test path for fulfilling the ***all c-uses*** criterion for the `sum` function. Please denote the required test path and mark this path in the control flow diagram.
e) Please determine the minimal necessary test path for fulfilling the ***all p-uses*** criterion for the `sum` function. Please denote the required test path and mark this path in the control flow diagram.
f) Please determine the minimal necessary test path for fulfilling the ***all c-uses/some p-uses*** criterion for the `sum` function. Please denote the required test path and mark this path in the control flow diagram.

Hint:
the path to a c-use ends in the node of the use,
p-uses have two paths (one for each path of the decision)

Example:

use	Defined in	Path	Variable
p1	n1	n1,n2,n4	i
p2	n1	n1,n2,n3	i
c1	n1	n1,n2,n3,n5	j

Problem 2: Path Coverage Test

a) Please determine the minimal necessary test cases for fulfilling the structured path coverage test for the parameter ***k=1*** for the `sum` operation.

b) Please determine the minimal necessary test cases for fulfilling the ***boundary interior test*** for the `sum` operation.

Problem 3: Equivalence Class Partitioning with Boundary Value Analysis

A student data management program processes registration number, name, major, and mark (on average) of every single student. The student registration number is a five-digit integer that is not smaller than 10000. The program knows the majors Mathematics, Computer Science, Philosophy, and English. The program knows the marks 1.0, 2.0, 3.0, 4.0, and 5.0. A valid name has at least 3 and at most 20 characters.

a) Please determine the equivalence class partitions for the given functionalities.

b) Please determine the test cases for all the equivalence classes using the procedure of boundary value analysis.

Problem 4: State-based Test

Given is the specification of a digital watch software.
For adjustment of a digital watch, the following states are to be considered:

- *Normal time:* State after inserting the battery
- *Adjust Hours:* Hours can be adjusted
- *Adjust Minutes:* Minutes can be adjusted
- *Adjust Seconds:* Seconds can be adjusted

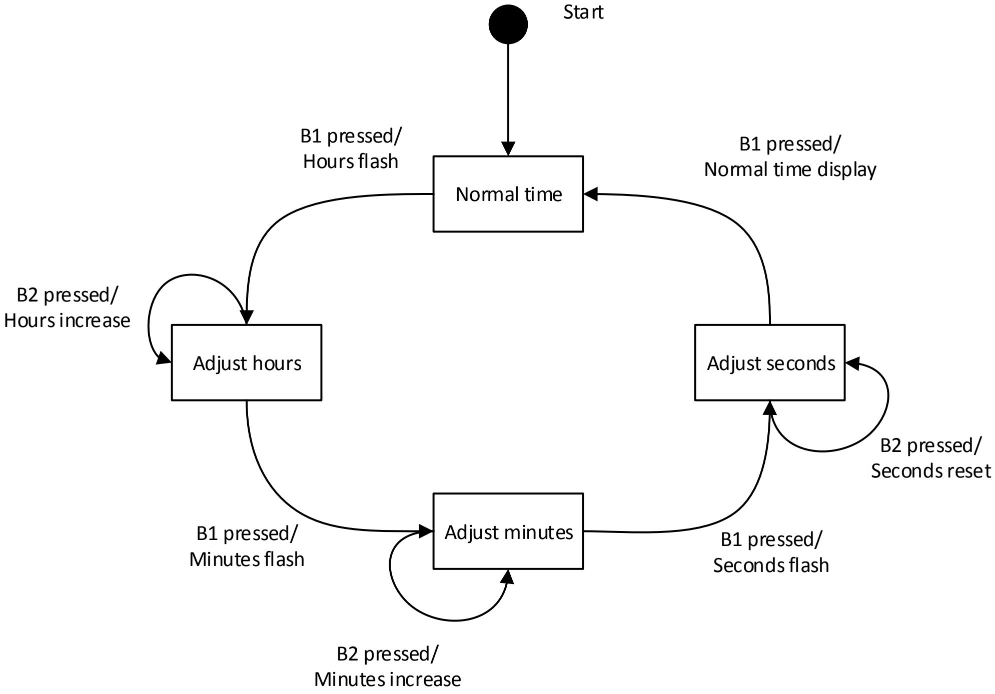
The following events could occur:

- *Start signal:* Battery inserted
- *Button 1 pressed*
- *Button 2 pressed*
- The two buttons must not be pressed simultaneously.

The following outputs could happen:

- *Hours flash:* The operator is currently in the hour editing mode.
- *Minutes flash:* The operator is currently in the minute editing mode.
- *Seconds flash:* The operator is currently in the second editing mode.
- *Hours increase:* The hour display has increased by 1 hour.
- *Minutes increase:* The minutes display increases by 1 minute.
- *Seconds reset:* 00 displays as second display.
- *Initialization:* Display of 00:00:00

State chart “Watch adjustment“



a) Please determine the test data for the program execution that traverses every state. Please select the simplest test cases.

b) Please determine the test data for the program execution that traverses every transition. Please select the simplest test cases.