

Assignment 1

ED. KAZMIERCZAK

SECOND SEMESTER, 2003

Introduction

The first assignment deals with black box testing. You are given the program `shuffle.c` that implements a simple (and somewhat constrained) card shuffling and card dealing “*module*”. The code and the specification for the shuffle program can be down-loaded from the subject web-page. It is also attached as an appendix to the assignment sheet.

The aim of this assignment is to explore the shuffle program using “*black-box*” testing strategies. Further, since you also have the source code for the shuffle program you can trace through the paths of the program exercised by your test cases, and determine for yourself the effectiveness of your choice of test cases.

Of course, there are faults in the program and you can probably uncover most of them by inspection. The point is to build up your own intuition.

Note: The program is written in C (an imperative language). We have not used an object oriented language because inheritance, polymorphism and (especially) dynamic binding complicate the testing process¹.

Tasks and Questions

Task 1

Your first task is to down-load the program and to make sure that you can compile it.

Question 2

What is the input domain for the program? What is the subset of *valid* inputs and what is the subset of *invalid* inputs?

Question 3

What are the equivalence classes for the input domain of the shuffle program? Does your set of equivalence classes cover the input space?

¹But we will explore testing of object oriented programs later in the semester!

Question 4

What are the test cases associated with the your equivalence classes?

Clearly specify your test cases in terms of inputs an expected outputs. Execute your test cases and record the results.

Question 5

Conduct a boundary value analysis for the shuffle program. What are your boundaries?

Specify the test cases associated with your boundaries. Execute the cases that are different to the test cases obtained by equivalence partitioning.

Question 6

Compare the two sets of test cases. Which is method is more effective and why? You should consider the test cases that you selected and which aspects of the input domain and output domain those test cases exercised.

But: please keep your comparison brief! No more than one page please!

The Shuffle Program Specification

The aim of the shuffle program is to create a module that shuffles and deals cards. The aim is to simulate real card shuffling. Recall that a standard pack of cards has 52 cards.

1. The user will be required to input two integers into the program. One integer specifies the number of players that require cards, and the other integer specifies the number of cards that each player receives.
2. The module must be able to deal cards to a minimum of 2 players and a maximum of 10 players.
3. The module must be able to deal out a subset of the cards from the deck or the entire deck to the players.
4. The set of cards dealt to a player must be random, that is, the probability of receiving the same cards on two consecutive deals is very low.
5. Each player receives the same number of cards.
6. The cards that each player receives must be unique, that is, the sets of cards that a player receives must be disjoint from the cards that any other player receives.
7. A player must receive exactly the number of cards specified.
8. The output of the program must list all of the players and the cards that are dealt to each of the players.