



Software Testing

Assignment: 2

Submitted To: Mr. Samir Obaid

Group Members

Name	Registration No.
Raja Hamza Zahoor	BSE173099
Warisha Waseem	BSE173021
Abdullah Naseer	BSE173048

Contents

1. Case Study	3
2. Functions.....	4
Function 1	4
Function 2	4
Function 3	4
3. Blackbox Testing.....	4
3.1 Worst BVA = 5^n.....	4
Function 1	4
Function 2	5
Function 3	6
3.2 Strong robust equivalence classes.....	6
Function 1	6
Function 2	7
Function 3	8
3.3 Comparison of Strong robust equivalence classes and robust worst case BVA	8
Function 1 = int checkYourCategory(int age, float income, float totalWealth).....	8
Function 2 = int accountAliveFor(int category, float amount)	8
Function 3 = int withdrawLimit(int category, float amount)	8

Prime Minister Ehsaas Program

1. Case Study

1.1 Introduction

Prime Minister Imran Khan has started Ehsaas Program to support needy and poor people in Pakistan. The government started sending 12000 cash to those who registered themselves in ehsaas program in Pakistan. Ehsaas is about the creation of a 'welfare state' by countering elite capture and leveraging 21st century tools—such as using data and technology to create precision safety nets; promoting financial inclusion and access to digital services; supporting the economic empowerment of women; focusing on the central role of human capital formation for poverty eradication, economic growth and sustainable development; and overcoming financial barriers to accessing health and post-secondary education. The program is for the extreme poor, orphans, widows, the homeless, the disabled, those who risk medical impoverishment, for the jobless, for poor farmers, for laborers, for the sick and undernourished; for students from low-income backgrounds and for poor women and elderly citizens. This plan is also about lifting lagging areas where poverty is higher.

1.2 Description

Individuals can register themselves using Ehsaas Program Portal. The portal allows the individual to check their eligibility. The cash which will be granted to individuals depends on their age and income. Different amounts are granted depending upon Age and Income factor. However the grant also depends upon age limit which is minimum 55 years old and maximum 85 years old. Individuals whose age is less or more than defined limit are not eligible for grant. System divides individuals' accounts into categories on which the time duration for how much time their account will be active/alive depends upon. System allows user to check balance. If balance is less than 4000 his/her account will be active for 50 days and if amount is greater than 4000 account will be active for 30 days. User must have to withdraw money within defined time limit to keep the system aware that he/she needs money. Otherwise money will be credited back to government on no-utilization money factor and account will be closed. Individual can withdraw money from system according to their category.

2. Functions

Function 1

`int checkYourCategory(int age, float income, float totalWealth)`

This function has 3 parameters of age, income and total wealth of a person on which this function calculates that whether a particular person is eligible of this program or not.

Function 2

`int accountAliveFor(int category, float amount)`

This function has 2 parameters of category and amount. On bases of these parameters it shows that how for how many days a specific account is active, after which the account is suspended by authorities.

Function 3

`int withdrawLimit(int category, float amount)`

This function has 2 parameters of category and amount. On bases of these parameters it shows the amount of time required between two transitions for a particular account.

3. Blackbox Testing

3.1 Worst BVA = 5^n

Function 1

`int checkYourCategory(int age, float income, float totalWealth)`

Total test cases = $5^3 = 125$

We do 63 in this assignment

The input values for this functions are:

Age = 55, 56, 70, 84, 85

Income = 0, 1, 3500, 6999, 7000

Total Wealth = 0, 1, 250,000, 499,999, 500,000

Sr.	Age	Income	Total Wealth	Category
1.	55	0	0	1
2.	55	0	1	1
3.	55	0	250000	2
4.	55	0	49	2
5.	55	0	50	2
6.	55	1	0	1
7.	55	1	1	1
8.	55	1	250000	2
9.	55	1	49	2
10.	55	1	50	2
11.	55	3500	0	1
12.	55	3500	1	1
13.	55	3500	250000	3
14.	55	3500	49	3
15.	55	3500	50	3
16.	55	6999	0	1
17.	55	6999	1	1
18.	55	6999	250000	3
19.	55	6999	49	3

20.	55	6999	50	3
21.	55	7000	0	1
22.	55	7000	1	1
23.	55	7000	250000	3
24.	55	7000	49	3
25.	55	7000	50	3
26.	56	0	0	1
27.	56	0	1	1
28.	56	0	250000	2
29.	56	0	49	2
30.	56	0	50	2
31.	56	1	0	1
32.	56	1	1	1
33.	56	1	250000	2
34.	56	1	49	2
35.	56	1	50	2
36.	56	3500	0	1
37.	56	3500	1	1
38.	56	3500	250000	3
39.	56	3500	49	3
40.	56	3500	50	3
41.	56	6999	0	1
42.	56	6999	1	1
43.	56	6999	250000	3
44.	56	6999	49	3
45.	56	6999	50	3
46.	56	7000	0	1
47.	56	7000	1	1
48.	56	7000	250000	3
49.	56	7000	49	3
50.	56	7000	50	3
51.	70	0	0	1
52.	70	0	1	1
53.	70	0	250000	1
54.	70	0	49	1
55.	70	0	50	1
56.	70	1	0	1
57.	70	1	1	1
58.	70	1	250000	1
59.	70	1	49	1
60.	70	1	50	1
61.	70	3500	0	1
62.	70	3500	1	1
63.	70	3500	250000	1

Function 2

`int accountAliveFor(int category, float amount)`

Total test cases = $5^2 = 25$

The input values for this functions are:

category = 1, 2, 3 (min+1= 2, max – 1= 2)

Amount = 0, 1, 6,000, 11,999, 12,000

Sr.	Category	Amount	Account is alive for days
1.	1	0	60
2.	1	1	60
3.	1	6,000	30
4.	1	11,999	30
5.	1	12,000	30
6.	2	0	30
7.	2	1	30
8.	2	6,000	20
9.	2	11,999	20
10.	2	12,000	20
11.	3	0	20
12.	3	1	20
13.	3	6,000	10
14.	3	11,999	10
15.	3	12,000	10

Function 3

`int withdrawLimit(int category, float amount)`

Total test cases = $5^2 = 25$

The input values for this functions are:

category = 1, 2, 3 (min+1= 2, max – 1= 2)

Amount = 0, 1, 6,000, 11,999, 12,000

Sr.	Category	Amount withdrawn	Next Withdraw in hour(s)
1.	1	0	0
2.	1	1	0
3.	1	6,000	0
4.	1	11,999	0
5.	1	12,000	0
6.	2	0	1
7.	2	1	1
8.	2	6,000	5
9.	2	11,999	Invalid
10.	2	12,000	Invalid
11.	3	0	2
12.	3	1	2
13.	3	6,000	24
14.	3	11,999	Invalid
15.	3	12,000	invalid

3.2 Strong robust equivalence classes

Function 1

`int checkYourCategory(int age, float income, float totalWealth)`

The input values for this functions are:

Age = 55 – 85 (we use 50, 65, 90)

Income = 0 – 7000 (we use -2, 4000, 8000)

Total Wealth = 0 - 500,000 (we use -2, 300,000, 800,000)

Sr.	Age	Income	Total Wealth	Category
1.	50	-2	-2	Invalid
2.	50	-2	300,000	Invalid
3.	50	-2	800,000	Invalid
4.	50	4000	-2	invalid
5.	50	4000	300,000	Invalid
6.	50	4000	800,000	Invalid
7.	50	8000	-2	Invalid
8.	50	8000	300,000	invalid
9.	50	8000	800,000	Invalid
10.	65	-2	-2	Invalid
11.	65	-2	300,000	Invalid
12.	65	-2	800,000	invalid
13.	65	4000	-2	Invalid
14.	65	4000	300,000	3
15.	65	4000	800,000	Invalid
16.	65	8000	-2	Invalid
17.	65	8000	300,000	Invalid
18.	65	8000	800,000	Invalid
19.	90	-2	-2	Invalid
20.	90	-2	300,000	Invalid
21.	90	-2	800,000	Invalid
22.	90	4000	-2	invalid
23.	90	4000	300,000	Invalid
24.	90	4000	800,000	Invalid
25.	90	8000	-2	Invalid
26.	90	8000	300,000	Invalid
27.	90	8000	800,000	invalid

Function 2

`int` accountAliveFor(`int` category, `float` amount)

The input values for this functions are:

category = 1 – 3 (we use 0, 2, 4)

Amount = 0 - 12,000 (we use -2, 6000, 14000)

Sr.	Category	Amount	Account is alive for days
1.	0	-2	Invalid
2.	0	6,000	Invalid
3.	0	14,000	Invalid
4.	2	-2	Invalid
5.	2	6,000	20
6.	2	14,000	Invalid
7.	4	-2	Invalid
8.	4	6,000	Invalid
9.	4	14,000	Invalid

Function 3

`int withdrawLimit(int category, float amount)`

The input values for this functions are:

category = 1 - 3 (we use 0, 2, 4)

Amount = 0 - 12,000 (we use -2, 6000, 14000)

Sr.	Category	Amount withdrawn	Next Withdraw in hour(s)
1.	0	-2	Invalid
2.	0	6,000	Invalid
3.	0	14,000	Invalid
4.	2	-2	Invalid
5.	2	6,000	Invalid
6.	2	14,000	Invalid
7.	4	-2	Invalid
8.	4	6,000	5
9.	4	14,000	Invalid

3.3 Comparison of Strong robust equivalence classes and robust worst case BVA

Function 1 = `int checkYourCategory(int age, float income, float totalWealth)`

In this fun the strong robust equivalence class has 27 test cases and robust worst case BVA has:

$7^3 = 343$ test cases.

Function 2 = `int accountAliveFor(int category, float amount)`

In this fun the strong robust equivalence class has 9 test cases and robust worst case BVA has:

$7^2 = 49$ test cases.

Function 3 = `int withdrawLimit(int category, float amount)`

In this fun the strong robust equivalence class has 27 test cases and robust worst case BVA has:

$7^2 = 49$ test cases.