Table of Contents

1. Case Study

There is a shop in a city where people buy and sale the computers. There is a huge stock of computers available. A person can only buy a computer if his age is greater than or equal to 20 and the age less than or equal to 40, he must have amount greater than or equal to 20000 and amount less than or equal to 40000 and he must has experience of using computer greater than or equal to 0 years and experience less than or equal to 15 years. Similarly the customer can sale his/her computer. He/she can also repair his/her computer.

2. Functions

- Bool isBuy(int age ,int anount, int experience) //returns answer yes or no …required function
  - 20<=age<=40
  - 20000<=amount<=40000
  - 5<=experience<=15

- Bool isSale(int age , int experience)

- Bool isRepaire(int age , int experience)

3. Worst Case BVA

Worst case will make all possible combination of 5 values from each set

- age = {20,21,25,39,40}
- amount = {20000,20001,30000,39999,40000}
- experience= {5,6,10,14,15}

5^n => 5^3 =125

4. Worse Test Cases

case	age	amount	experience	output
1	20	20000	5	yes
2	20	20000	21	yes
3	20	20000	25	yes
4	20	20000	39	yes
5	20	20000	40	yes
6	20	20001	5	yes
7	20	20001	21	yes

8	20	20001	25	yes
9	20	20001	39	yes
10	20	20001	40	yes
11	20	30000	5	yes
12	20	30000	21	yes
13	20	30000	25	yes
14	20	30000	39	yes
15	20	30000	40	yes
16	20	39999	5	yes
17	20	39999	21	yes
18	20	39999	25	yes
19	20	39999	39	yes
20	20	39999	40	yes
21	21	40000	5	yes
22	21	40000	21	yes
23	21	40000	25	yes
24	21	40000	39	yes
25	21	40000	40	yes
26	21	20000	5	yes
27	21	20000	21	yes
28	21	20000	25	yes
29	21	20000	39	yes
30	21	20000	40	yes
31	21	20001	5	yes
32	21	20001	21	yes
33	21	20001	25	yes
34	21	20001	39	yes
35	21	20001	40	yes
36	21	30000	5	yes
37	21	30000	21	yes
38	21	30000	25	yes
39	21	30000	39	yes
40	21	30000	40	yes
41	25	39999	5	yes
42	25	39999	21	yes
43	25	39999	25	yes
44	25	39999	39	yes
45	25	39999	40	yes
46	25	40000	5	yes
47	25	40000	21	yes
48	25	40000	25	yes
49	25	40000	39	yes
50	25	40000	40	yes
51	25	20000	5	yes
52	25	20000	21	yes

53	25	20000	25	yes
54	25	20000	39	yes
55	25	20000	40	yes
56	25	20001	5	yes
57	25	20001	21	yes
58	25	20001	25	yes
59	25	20001	39	yes
60	25	20001	40	yes

- **Strong robust equivalence class partitioning**

1. **Bool isBuy(int age ,int anount, int experience) //returns answer yes or no function**
    - 20<=age<=40
    - 20000<=amount<=40000
    - 0<=experience<=20
    - **Classes // changes done after receiving feedback**
      Consider age as: 20 to 23, 24 to 27, 28 to 31, 32 to 35, and 36 to 40
      Consider amount as: 20000 to 23000, 24000 to 27000, 28000 to 31000, 32000 to 35000, and 36000 to 40000
      Consider experience as: 0 to 2, 3 to 5, 6 to 8, 10 to 12, and 13 to 15

    - **Test Cases for Bool isBuy(int age ,int anount, int experience)**

case	age	amount	experience	output
1	19	19000	-1	yes
2	19	19000	1	yes
3	19	19000	3	yes
4	19	19000	6	yes
5	19	19000	10	yes
6	19	19000	13	yes
7	19	19000	16	yes
8	19	21000	-1	yes
9	19	21000	1	yes
10	19	21000	3	yes
11	19	21000	6	yes
12	19	21000	10	yes
13	19	21000	13	yes
14	19	21000	16	yes
15	19	25000	-1	yes
16	19	25000	1	yes
17	19	25000	3	yes
18	19	25000	6	yes
19	19	25000	10	yes
20	19	25000	13	yes

21	19	25000	16	yes
22	19	29000	-1	yes
23	19	29000	1	yes
24	19	29000	3	yes
25	19	29000	6	yes
26	19	29000	10	yes
27	19	29000	13	yes
28	19	29000	16	yes
29	19	330000	-1	yes
30	19	330000	1	yes
31	19	330000	3	yes
32	19	330000	6	yes
33	19	330000	10	yes
34	19	330000	13	yes
35	19	330000	16	yes
36	19	37000	-1	yes
37	19	37000	1	yes
38	19	37000	3	yes
39	19	37000	6	yes
40	19	37000	10	yes
41	19	37000	13	yes
42	19	37000	16	yes
43	19	41000	-1	yes
44	19	41000	1	yes
45	19	41000	3	yes
46	19	41000	6	yes
47	19	41000	10	yes
48	19	41000	13	yes
49	19	41000	16	yes
50	20	19000	-1	yes
51	20	19000	1	yes
52	20	19000	3	yes
53	20	19000	6	yes
54	20	19000	10	yes
55	20	19000	13	yes
56	20	19000	16	yes
57	20	21000	-1	yes
58	20	21000	1	yes
59	20	21000	3	yes
60	20	21000	6	yes
61	20	21000	10	yes
62	20	21000	13	yes
63	20	21000	16	yes
64	20	25000	-1	yes
65	20	25000	1	yes

66	20	25000	3	yes
67	20	25000	6	yes
68	20	25000	10	yes
69	20	25000	13	yes
70	20	25000	16	yes
71	20	29000	-1	yes
72	20	29000	1	yes
73	20	29000	3	yes
74	20	29000	6	yes
75	20	29000	10	yes
76	20	29000	13	yes
77	20	29000	16	yes
78	20	330000	-1	yes
79	20	330000	1	yes
80	20	330000	3	yes
81	20	330000	6	yes
82	20	330000	10	yes
83	20	330000	13	yes
84	20	330000	16	yes
85	20	37000	-1	yes
86	20	37000	1	yes
87	20	37000	3	yes
88	20	37000	6	yes
89	20	37000	10	yes
90	20	37000	13	yes
91	20	37000	16	yes
92	20	41000	-1	yes
93	20	41000	1	yes
94	20	41000	3	yes
95	20	41000	6	yes
96	20	41000	10	yes
97	20	41000	13	yes
98	20	41000	16	yes

2. **Bool isSale(int age , int experience) //returns answer yes or no function**

    - **Classes // changes done after receiving feedback**

    - Consider age as: 20 to 23, 24 to 27, 28 to 31, 32 to 35, and 36 to 40
    - Consider experience as: 0 to 2, 3 to 5, 6 to 8, 10 to 12, and 13 to 15

- **Test cases for Bool isSale(int age , int experience)**

case	age	experience	output
1	19	-1	yes
2	19	1	yes
3	19	3	yes
4	19	6	yes
5	19	10	yes
6	19	13	yes
7	19	16	yes
8	21	-1	yes
9	21	1	yes
10	21	3	yes
11	21	6	yes
12	21	10	yes
13	21	13	yes
14	21	16	yes
15	25	-1	yes
16	25	1	yes
17	25	3	yes
18	25	6	yes
19	25	10	yes
20	25	13	yes
21	25	16	yes
22	29	-1	yes
23	29	1	yes
24	29	3	yes
25	29	6	yes
26	29	10	yes
27	29	13	yes
28	29	16	yes

29	33	-1	yes
30	33	1	yes
31	33	3	yes
32	33	6	yes
33	33	10	yes
34	33	13	yes
35	33	16	yes
36	37	-1	yes
37	37	1	yes
38	37	3	yes
39	37	6	yes
40	37	10	yes
41	37	13	yes
42	37	16	yes
43	41	-1	yes
44	41	1	yes
45	41	3	yes
46	41	6	yes
47	41	10	yes
48	41	13	yes
49	41	16	yes

3. **Bool isRepaire(int age , int experience) //returns answer yes or no function**

- **Classes // changes done after receiving feedback**

➢ Consider age as: 20 to 23, 24 to 27, 28 to 31, 32 to 35, and 36 to 40

➢ Consider experience as: 0 to 2, 3 to 5, 6 to 8, 10 to 12, and 13 to 15

- **Test Cases for Bool isRepaire(int age , int experience)**

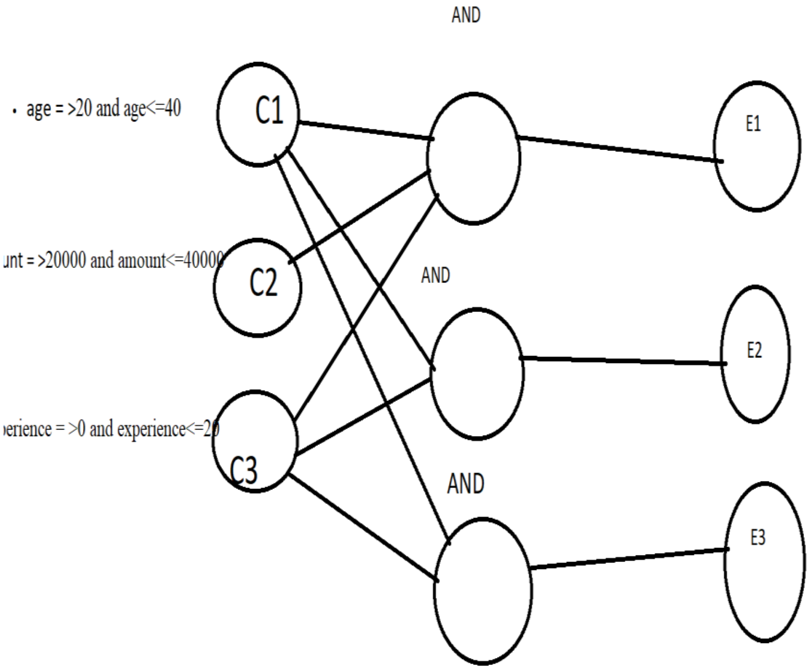
case	age	experience	output
1	19	-1	yes
2	19	1	yes
3	19	3	yes
4	19	6	yes
5	19	10	yes
6	19	13	yes
7	19	16	yes
8	21	-1	yes
9	21	1	yes
10	21	3	yes
11	21	6	yes
12	21	10	yes
13	21	13	yes
14	21	16	yes
15	25	-1	yes
16	25	1	yes
17	25	3	yes
18	25	6	yes
19	25	10	yes
20	25	13	yes
21	25	16	yes
22	29	-1	yes
23	29	1	yes
24	29	3	yes

25	29	6	yes
26	29	10	yes
27	29	13	yes
28	29	16	yes
29	33	-1	yes
30	33	1	yes
31	33	3	yes
32	33	6	yes
33	33	10	yes
34	33	13	yes
35	33	16	yes
36	37	-1	yes
37	37	1	yes
38	37	3	yes
39	37	6	yes
40	37	10	yes
41	37	13	yes
42	37	16	yes
43	41	-1	yes
44	41	1	yes
45	41	3	yes
46	41	6	yes
47	41	10	yes
48	41	13	yes
49	41	16	yes

a. List down requirements in form of causes and effects

- C1 : age = >20 and age<=40
- C2 : amount = >20000 and amount<=40000
- C3: experience = >0 and experience<=20
- E1: Yes (buys computer)
- E2: Yes (sales computer)
- E3: Yes (repairs computer)

b. Draw cause effect graphs


AND
age = >20 and age<=40
C1
E1
unt = >20000 and amount<=40000
C2
AND
E2
erience = >0 and experience<=20
C3
AND
E3

c. Decision table

		1	2
Condition / Cause	**C1** age = >20 and age<=40	1	1

Condition / Cause	**C2** amount = >20000 and amount<=40000	1	0
Condition / Cause	**C3** experience = >0 and experience<=20	1	1
Action / Effect	E1	X	-
Action / Effect	E2	-	X
Action / Effect	E3	-	X
Action / Effect	E4	-	-

d. Identify test cases

- Since there are 2 rules in our decision table above, so we must have at least 2 test cases to test this system using this technique.

These test cases can be

1. Age = 25, amount = 25000, experience =10.
1. Age = 27, experience =17.

e. Draw a table to mention test case number, test data and expected output

- **Boundary Value Analysis**

case	age	amount	experience	output
1	30	30000	0	Yes
2	30	30000	1	Yes
3	30	30000	10	Yes

4	30	30000	19	Yes
5	30	30000	20	Yes
6	30	20000	10	Yes
7	30	20001	10	Yes
8	30	39999	10	Yes
9	30	40000	10	Yes
10	20	30000	10	Yes
11	21	30000	10	Yes
12	39	30000	10	Yes
13	40	30000	10	Yes