Software Testing Methodologies: WASE

Sample exercises (with solutions)

Topic #1: Boundary Value Analysis

Q1. An image can be represented in the form of pixels. Each pixel has a [RGB] value which represents the Red, Green and Blue color components of the pixel. Each of the [RGB] values lie in the range 0-255 (end points included). A program takes the [RGB] value of a pixel and returns the most dominant color of the pixel (which ever component has the highest value is the dominant color).

a) Calculate the number of test cases – Normal BVA, Robust BVA, Worst case BVA and Robust worst case BVA.

For BVA :
Total number of test cases: 4 n + 1 = 4 * 3 + 1 = 13

For Robust BVA :
Total number of test cases: 6 n + 1 = 6 * 3 + 1 = 19

For Worst Case BVA :
Total number of test cases: 5^n= 5 * 5 * 5 = 125

For Robust Worst Case BVA :
Total number of test cases: 7^n= 7 * 7 * 7 = 343

b) Write all the test cases for Robust BVA for the pixel program.

Test Cases for Robust BVA

Test Case #	Red	Green	Blue	Expected Output
1	-1	100	200	Invalid
2	0	100	200	Blue
3	1	100	200	Blue
4	100	100	200	Blue
5	254	100	200	Red
6	255	100	200	Red
7	256	100	200	Invalid
8	100	-1	200	Invalid
9	100	0	200	Blue
10	100	1	200	Blue
11	100	254	200	Green
12	100	255	200	Green

13	100	256	200	Invalid
14	100	200	-1	Invalid
15	100	200	0	Green
16	100	200	1	Green
17	100	200	254	Blue
18	100	200	255	Blue
19	100	200	256	Invalid

Q2. An online banking portal allows users to login using a user name (5 char <= user name <= 12 characters long) and a registered password (4 char <= password <= 9 characters long).

a) Calculate the number of test cases – Normal BVA, Robust BVA, Worst case BVA and Robust worst case BVA required to carry out boundary-value analysis for the user name and password variables for the login page.

For BVA :
Total number of test cases: 4 n + 1 = 4 * 2 + 1 = 9

For Robust BVA :
Total number of test cases: 6 n + 1 = 6 * 2 + 1 = 13

For Worst Case BVA :
Total number of test cases: 5^n= 5 * 5 = 25

For Robust Worst Case BVA :
Total number of test cases: 7^n= 7 * 7 = 49

b) Write test cases (atleast 8) for Robust worst case BVA for the login page. .

Test Cases for Robust worst case BVA

5 char <= user name <= 12 characters long
4 char <= password <= 9 characters

Test Case #	User name	password	Expected Output
1	abcde (min)	1234 (min)	Valid length of username and password
2	abcde (min)	12345 (min +)	Valid length of username and password
3	abcde (min)	1234567 (nom)	Valid length of username and password
4	abcde (min)	12345678 (max-)	Valid length of username and password
5	abcde (min)	123456789 (max)	Valid length of username and password
6	abcde (min)	123 (min-)	Invalid length of password 4 char <= password <= 9 characters
7	abcde (min)	1234567890 (max+)	Invalid length of password 4 char <= password <= 9 characters

8	abcdef (min+)	1234 (min)	Valid length of username and password
9	abcdef (min+)	12345 (min +)	Valid length of username and password
10	abcdef (min+)	1234567 (nom)	Valid length of username and password
11	abcdef (min+)	12345678 (max-)	Valid length of username and password
12	abcdef (min+)	123456789 (max)	Valid length of username and password
13	abcdef (min+)	123 (min-)	Invalid length of password 4 char <= password <= 9 characters
14	abcdef (min+)	1234567890 (max+)	Invalid length of password 4 char <= password <= 9 characters

---

Topic #2: Equivalence Class Partitioning

Q1. A bank hosts a program on its website that determines the maximum amount for which a credit card can be issued to a user. The maximum limit is based on user annual income and age. The following is the criteria applied:

Age	Annual Income	Credit Card Max Limit
31 <= Age <= 40	3 Lacs <= Income <= 5 Lacs	Rs. 50,000
31 <= Age <= 40	5 Lacs < Income <= 10 Lacs	Rs. 75,000
31 <= Age <= 40	10 Lacs < Income <= 15 Lacs	Rs. 1 Lac
31 <= Age <= 40	Income > 15 Lacs	Rs. 2 Lacs
40 < Age <= 50	3 Lacs <= Income <= 5 Lacs	Rs. 75,000
40 < Age <= 50	5 Lacs < Income <= 10 Lacs	Rs. 1 Lac
40 < Age <= 50	10 Lacs < Income <= 15 Lacs	Rs. 2 Lac
40 < Age <= 50	Income > 15 Lacs	Rs. 3 Lacs

a) Derive the valid and invalid sub-domains for the input variables. Justify any specific choice you make for the sub-domains.

Number of EC for age = 4 (2 valid and 2 invalid)

Number of EC for income = 5 (1 invalid + 4 valid) (only 1 invalid case because upper limit is open)

b) Calculate the number of test cases for Strong Robust Equivalence Class testing. State clearly the sub-domains chosen.

Ans:

Number of EC for age = 4 (2 valid and 2 invalid)

Number of EC for income = 5 (1 invalid + 4 valid) (only 1 invalid case because upper limit is open)

Number of test cases for Strong Robust = 4 * 5 = 20

c) Write down all the test cases for Strong Robust Equivalence Class testing. State clearly the sub-domains chosen.

Test Case #	Age	Income	Expected Output
1	35	1 Lac	Invalid Income
2	35	2 Lac	Rs. 50,000
3	35	7 Lac	Rs. 75,000
4	35	12 Lac	Rs. 1 Lac
5	35	17 Lac	Rs. 2 Lacs
6	45	1 Lac	Invalid Income
7	45	2 Lac	Rs. 75,000
8	45	7 Lac	Rs. 1 Lac
9	45	12 Lac	Rs. 2 Lac
10	45	17 Lac	Rs. 3 Lacs
11	60	1 Lac	Invalid Age Invalid Income
12	60	2 Lac	Invalid Age
13	60	7 Lac	Invalid Age
14	60	12 Lac	Invalid Age
15	60	17 Lac	Invalid Age
16	20	1 Lac	Invalid Age Invalid Income
17	20	2 Lac	Invalid Age
18	20	7 Lac	Invalid Age
19	20	12 Lac	Invalid Age
20	20	17 Lac	Invalid Age

d) Identify the equivalence classes for the valid output domain. Write down sample test cases for the same.

Test Case #	Age	Income	Expected Output
1	35	2 Lac	Rs. 50,000
2	35	7 Lac	Rs. 75,000
3	35	12 Lac	Rs. 1 Lac
4	35	17 Lac	Rs. 2 Lacs
5	45	2 Lac	Rs. 75,000
6	45	7 Lac	Rs. 1 Lac
7	45	12 Lac	Rs. 2 Lac
8	45	17 Lac	Rs. 3 Lacs

Q2. Electric Supply Corporation of India has developed an online GUI based monthly electricity bill payment system with the following criteria

a. User has to input a valid customer ID of 11 digits of all positive integers.
b. Payment can be made by valid Debit card (validity date should be => date of payment).
c. The card number should be 16 digits of all positive integers
d. The CVV number should be of 3 digits of all positive integers.

i) Identify the positive and negative domain.
ii) Write sample test cases for weak robust variant.

ANSWER:

Positive Domain

Customer ID	**Card Validity Date**	**Card No**	**CVV**
12345678912	Current date of payment	4008822412344321	123
	Future date - later than the date of payment		

Negative Domain

Customer ID	Card Validity Date	Card No	CVV
0	Earlier than payment date	0	0
-1234567891	Earlier than payment Date	4008	1
1234	Earlier than payment date	-400882241234432	-12

ii) Sample Test Cases for Weak Robust Equivalence Class Testing

Test Case ID	Customer ID	Card Validity Date	Card No	CVV	Expected Results
001	12345678912	Current Date of payment	4008822412344321	123	Bill Payment Successful
002	12345678912	Future date	4008822412344321	123	Bill Payment Successful
003	12345678912	Future date	4008822412344321	1	Bill Payment Failed
004	12345678912	Future date	4008822412344321	-12	Bill Payment Failed
005	12345678912	Future date	4008822412344321	0	Bill Payment Failed
006	-1234567891	Future date	4008822412344321	123	Bill Payment Failed
007	1234	Future date	4008822412344321	123	Bill Payment Failed
008	0	Future date	4008822412344321	123	Bill Payment Failed
009	12345678912	Future date	4008	123	Bill Payment Failed
010	12345678912	Future date	-400882241234432	123	Bill Payment Failed
011	12345678912	Future date	0	123	Bill Payment Failed
012	12345678912	Earlier than date of payment	4008822412344321	123	Bill Payment failed

Topic #3: Decision Tables

Q1. MIIT organizes the BIGBANG test in order to shortlist candidates for entry to its various foundation courses. If the student scores 90% or more marks, he/she gets admission to the 2nd level course. However, if the student scores 80% or more marks (but less than 90%), he/she gets admission to 1st level course. Students below 80% are not entitled for admission.

i) Identify the conditions and actions for the given case, in order to draw the decision table.
ii) Derive the test cases with the values of inputs and outputs.

ANSWER:

C1: marks >=90.
C2: 80 <=marks <90
C3: marks < 80
A1: Admission to 2nd level course
A2: Admission to 1st level course
A3: No admission
A4: Impossible

a) Derive the decision table for the problem.
Ans:

Stud	***Rule 1***	***Rule 2***	***Rule 3***	***Rule 4***	***Rule 5***	***Rule 6***	***Rule 7***	***Rule 8***
C1: marks >=90	T	T	T	T	F	F	F	F
C2: 80 <=marks <90	T	T	F	F	T	T	F	F
C3: marks < 80	T	F	T	F	T	F	T	F
A1: 2nd level				X				
A2: 1st level						X		
A3: No admission							X	
A4: Impossible	X	X	X		X			X

b) Design the test cases with values of inputs and outputs.

Test Case #	**Marks**	**Expected Output**
DT1	95	Admission to 2nd level
DT2	82	Admission to 1st level
DT3	73	No Admission

Q2. A marketing company wishes to construct a decision table to decide how to treat clients according to three characteristics: Gender, City Dweller, and age group: A (under 30), B (between 30 and 60), C (over 60). The company has four products (W, X, Y and Z) to test market. Product W will appeal to female city dwellers. Product X will appeal to young females. Product Y will appeal to Male middle aged shoppers who do not live in cities. Product Z will appeal to all but older females. Construct a decision table for this problem –

(i) Identify the Condition and action entry.
(ii) Create the decision table.

Answer:

	Characteristics	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
C1	Gender	F	M	F	M	F	F	M	F	F	M
C2	City Dweller	Y	Y	N	N	Y	N	N	Y	N	N
C3	Age Group	A	-	A	A	B	B	B	C	C	C
A1	W		X			X			X		
A2	X	X		X							
A3	Y							X			
A4	Z	X	X	X	X	X	X	X			X

Topic #4: Control Flow Testing

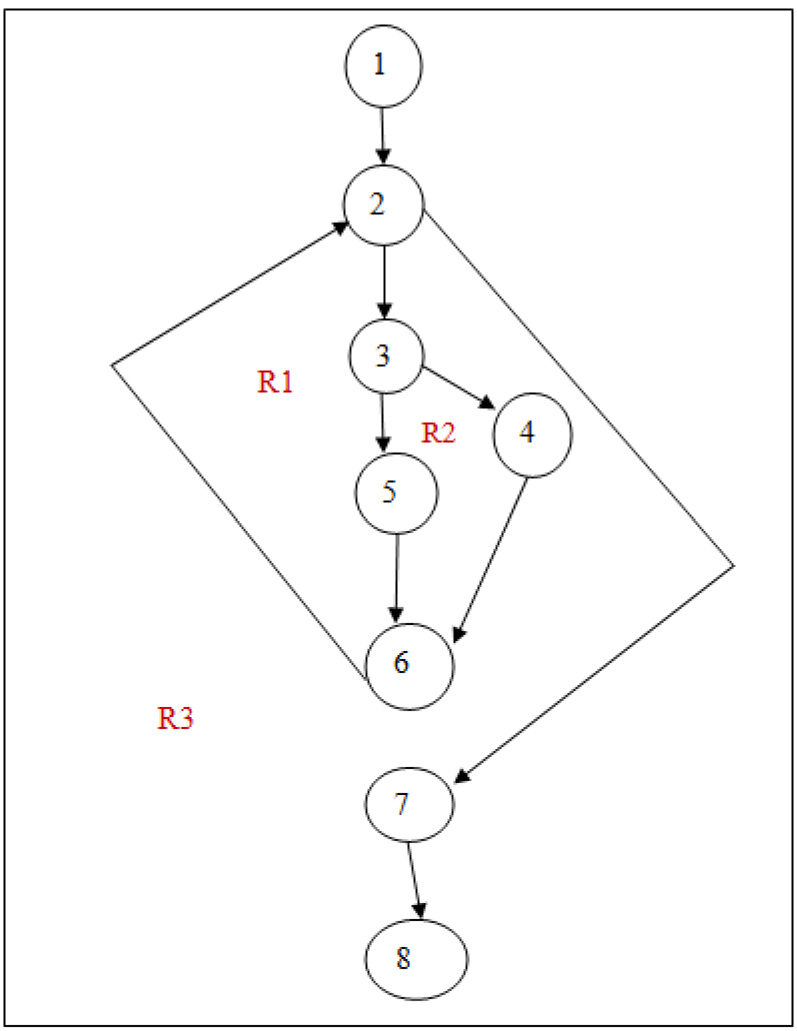
Q1. Consider the binary search algorithm given below:

```
algo binarySearch(arr, size, searchValue)
{
   int low =0;
   int high = size-1;
   int mid = (low + high) /2;
   int NOT_FOUND = -1;
   while (low < = high && arr[mid] != searchValue)
   {
        if (arr[mid] < searchValue) low = mid +1;
        else high = mid - 1;
        mid = (low + high)/2;
   }
   if (low > high) mid = NOT_FOUND;
   return mid;
}
```

Answer the following questions:

a) Draw the control flow graph for the given algorithm.

```
algo binarySearch(arr, size, searchValue)
{
int low =0;
int high = size-1;                          1
int mid = (low + high) /2;
int NOT_FOUND = -1;
while (low < = high && arr[mid] != searchValue)   2
{
  3  if (arr[mid] < searchValue)  low = mid +1;    4
     else high = mid - 1;
6    mid = (low + high)/2;                        5
}
if (low > high) mid = NOT_FOUND;   7
8 return mid;
}
```



a) Calculate Cyclomatic Complexity using all the three methods.

V(G) = P + 1= 2 + 1= 3 (Nodes 2 and 3 are predicate nodes)

V(G) = Number of regions = 3 (should be marked in the diagram)

V(G) = E – N +2 = 9-8 + 2= 1 + 2 =3

b) Find the basis set of all the execution paths.

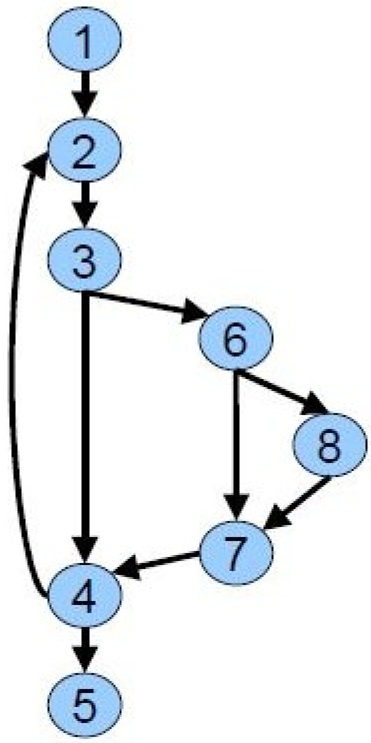
Basis Set:

Path 1: 1-2-7-8

Path 2: 1-2-3-5-6-2-7-8

Path 3: 1-2-3-4-6-2-7-8

Q2. Find the Cyclomatic complexity and basis paths of the given CFG.



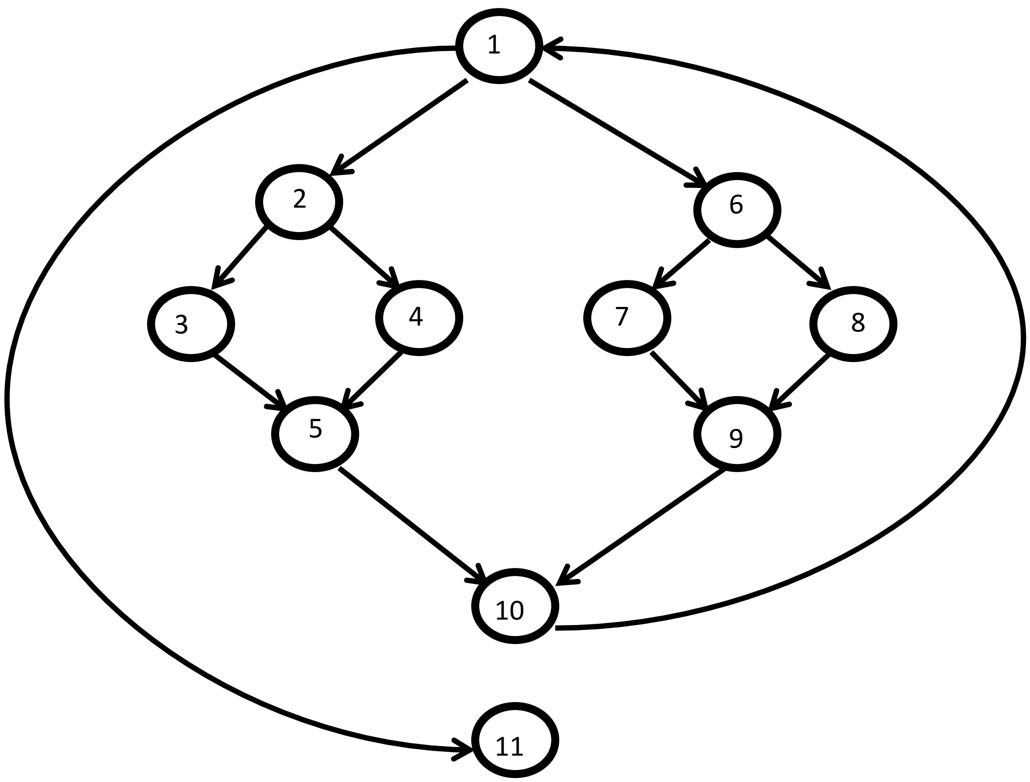
Answer Keys

- Cyclomatic Complexity
  - V(G) = e- n + 2(p) = 10 – 8 + 2 (1) = 4

- **Basis Paths**
  - Path 1 = 1-2-3-4-5
  - Path 2 = 1-2-3-4-2-3-4-5
  - Path 3 = 1-2-3-6-7-4-5
  - Path 4 = 1-2-3-6-8-7-4-5

Q3. Find the cyclomatic complexity and basis paths of the following control flow graph

(i) Cyclomatic Complexity
(ii) Basis Paths



ANSWER:

i) Cyclomatic Complexity V(G) = e – n + 2(p) = 14 – 11 + 2 (1) = 5

ii) Basis Paths

Path 1 = 1-2-3-5-10-1-11

Path 2 = 1-2-4-5-10-1-11

Path 3 = 1-6-7-9-10-1-11

Path 4 = 1-6-8-9-10-1-11

Path 5 = 1-11