CMR
INSTITUTE OF
TECHNOLOGY

USN □□□□□□□□□□

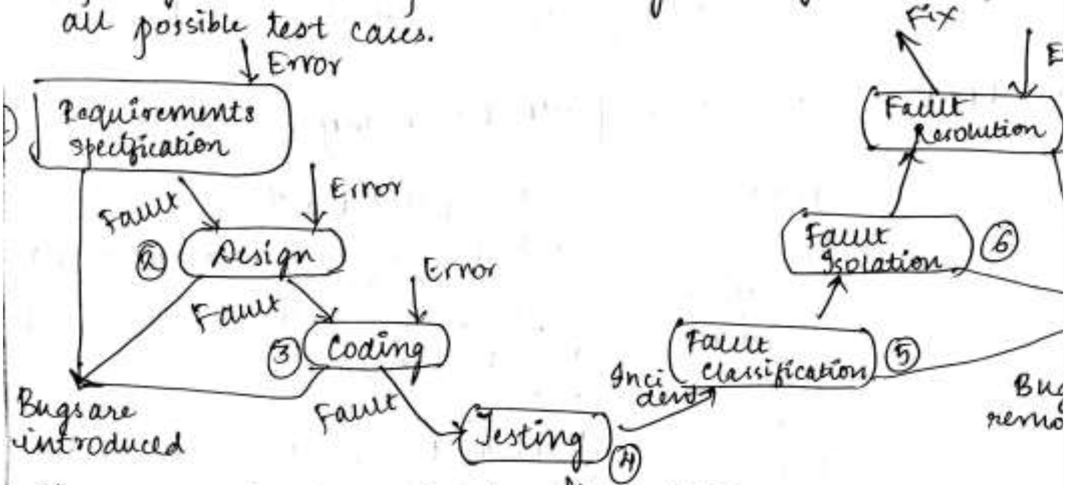
Internal Assessment 1 - Scheme and Solutions

Sub:	**SOFTWARE TESTING**							Code:	**10IS65**
Date:	30 / 03 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	**ISE**
Answer Any **FIVE FULL** Questions									

		Marks	OBE	
			CO	RBT
1 (a)	What is Software Testing? Why is it so important in SDLC?	[06]	CO1	L1

Software Testing: It is a process of identifying the completeness, correctness and quality of the developed computer software.

It is considered important is SDLC - software development life cycle as it helps in removing the bugs and verifies all possible test cases.



The SDLC is shown in the figure above.

There are 3 phases where errors can occur. ~~It~~. they are therequirement spec phase, design phase and coding phas

These errors get converted into faults and propogate throughou the process.

the development phases are phases ① ,② & ③.

These are the phases where the bugs are introduced.

As we can observe the test cases occupy the central position in the process.

Here in the testing phases all the ~~to~~ test cases are tested and bugs are examined.

(b)	Explain IEEE error and fault taxonomy.	[04]	CO1	L4

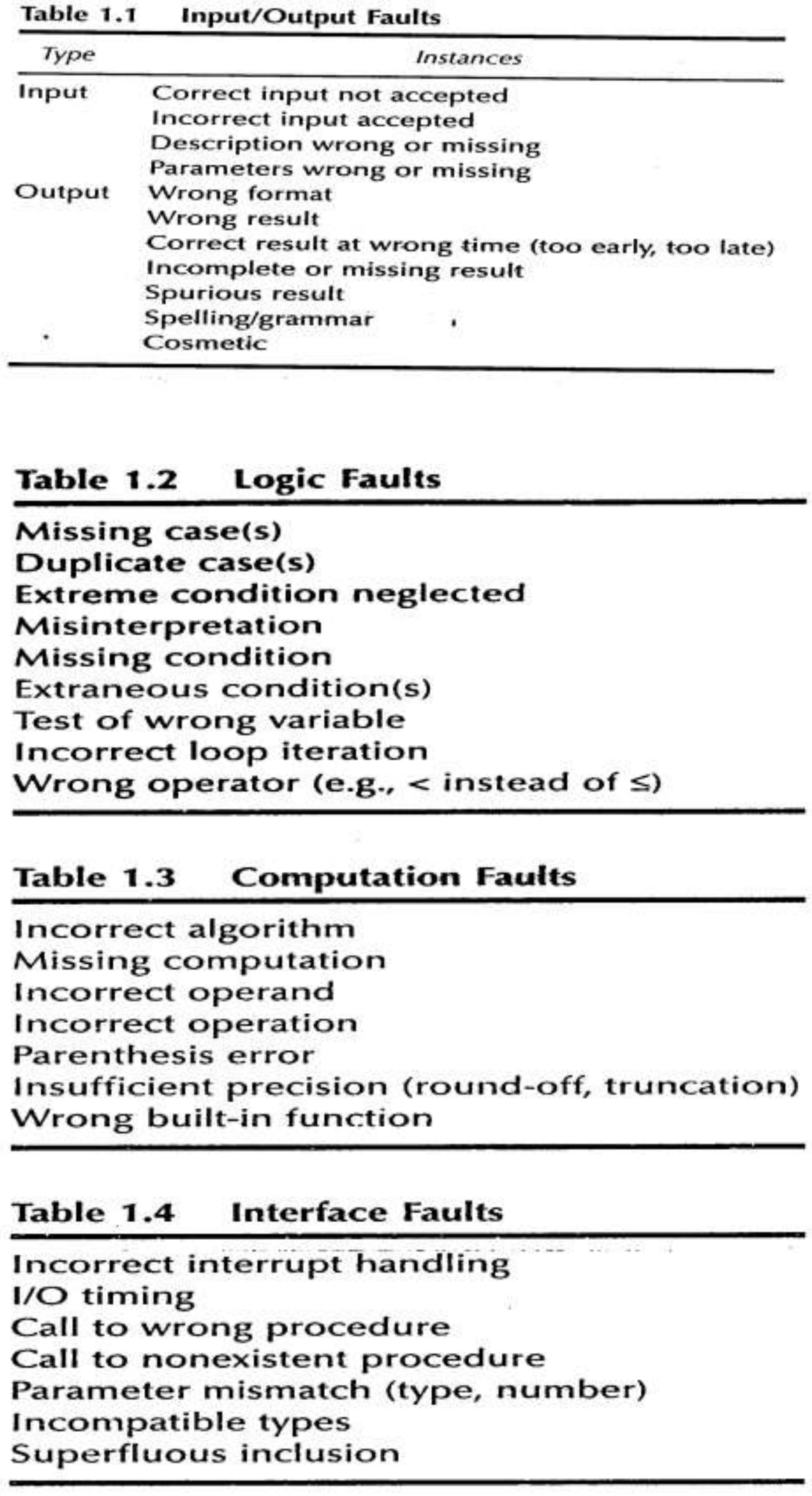


Table 1.1 Input/Output Faults

Type	Instances
Input	Correct input not accepted
	Incorrect input accepted
	Description wrong or missing
	Parameters wrong or missing
Output	Wrong format
	Wrong result
	Correct result at wrong time (too early, too late)
	Incomplete or missing result
	Spurious result
	Spelling/grammar
	Cosmetic

Table 1.2 Logic Faults

Missing case(s)
Duplicate case(s)
Extreme condition neglected
Misinterpretation
Missing condition
Extraneous condition(s)
Test of wrong variable
Incorrect loop iteration
Wrong operator (e.g., < instead of ≤)

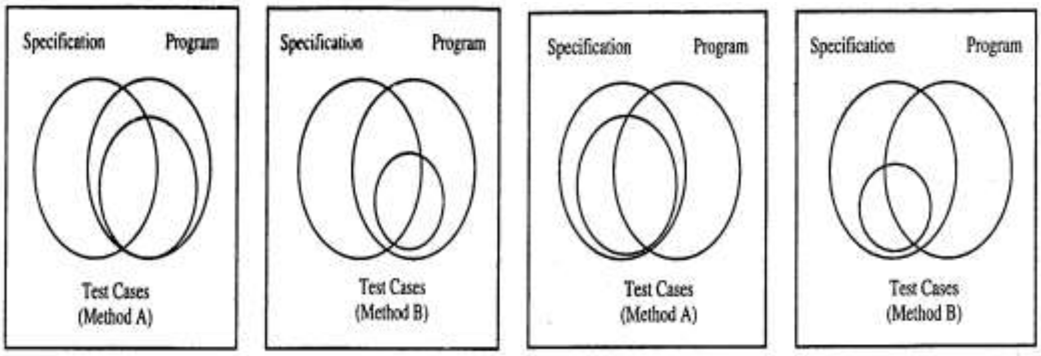
Table 1.3 Computation Faults

Incorrect algorithm
Missing computation
Incorrect operand
Incorrect operation
Parenthesis error
Insufficient precision (round-off, truncation)
Wrong built-in function

Table 1.4 Interface Faults

Incorrect interrupt handling
I/O timing
Call to wrong procedure
Call to nonexistent procedure
Parameter mismatch (type, number)
Incompatible types
Superfluous inclusion

2 (a)	Differentiate between Structural Testing and Functional Testing, explain using Venn diagram.	[06]	CO1	L2



Structural testing uses the help of Specifications in order to identify test cases

Functional testing uses the source code for the identification of test cases

None of the processes are inadequate and complete without each other

Certain program behaviours which are specificed but not implemented, structural testing will never be aware of these program behaviours

Certain program behaviours which will be implemented but will not be specified, ~~those~~ functional testing will never reveal such program behaviours

Both the testing types are very important

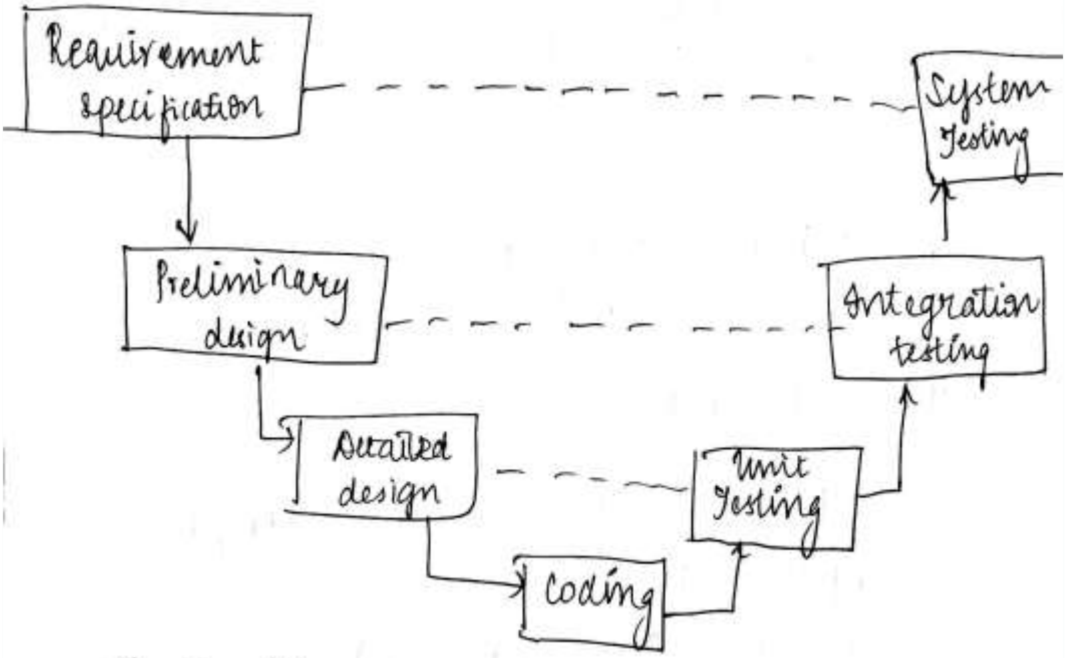
It helps in both recognizing and resolving faults

Structural testing is important is unit level

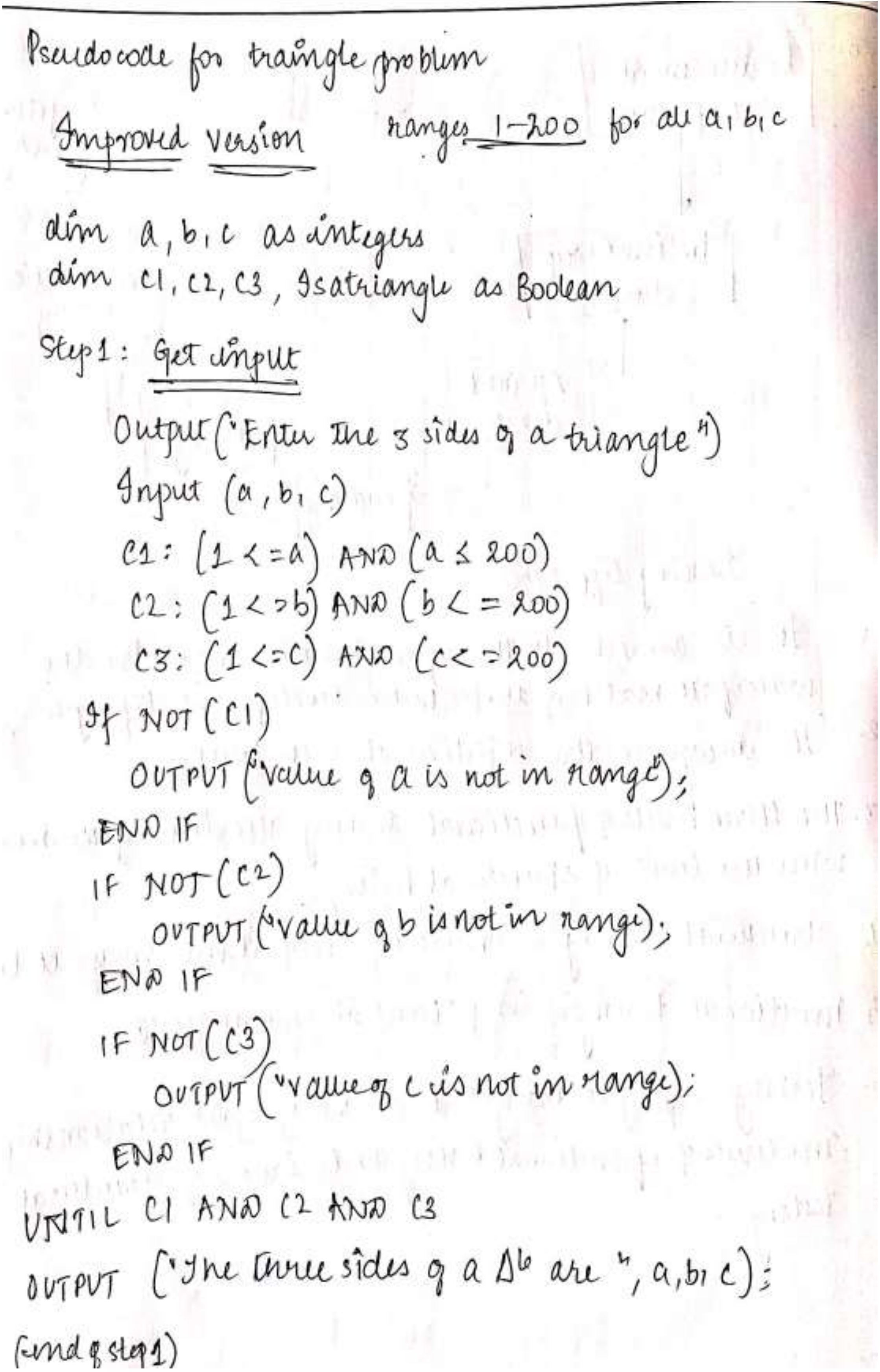
Functional testing is important as system level

Structural testing seeks faults

Functional testing establishes confidence.

(b)	Explain testing life cycle with a neat diagram.  Requirement specification System Testing Preliminary design Integration testing Detailed design Unit Testing Coding Testing lifecycle It is an echo to the approximations made in the waterfall model of the software development lifecycle . It recognises the objectives at each level The three levels of functional testing are directly associate with the levels of structural testing. Structural testing is considered important in unit lev Functional testing is important at system level Testing life cycle brings out a very good relationship b/ the levels of functional testing and levels of structural testing .	[04]	CO1	L5
3 (a)	Justify the usage of Boundary Value Analysis with function of two variable and perform Output BVA on Commission Problem.	[10]	CO2	L5

4 (a)	Discuss the pseudocode of Triangle Problem and perform decision table approach testing.	[10]	CO2	L2



Pseudocode for traingle problem

Improved version   ranges 1-200 for all a, b, c

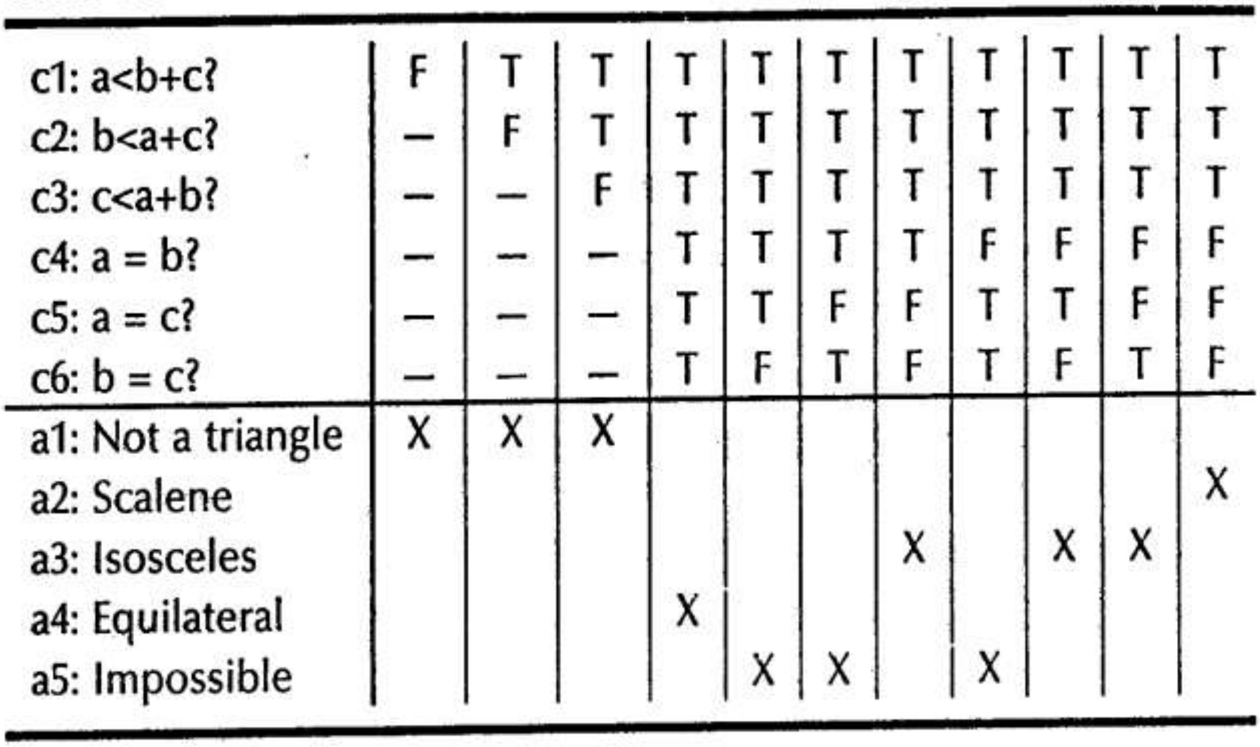
```
dim  a, b, c  as integers
dim  c1, c2, C3 , Isatriangle  as Boolean
Step 1 :  Get input
      Output ("Enter The 3 sides of a triangle")
      Input (a , b, c)
      C1 : (1 <= a) AND (a ≤ 200)
      C2 : (1 <>b) AND (b <= 200)
      C3 : (1 <=C) AND (c<=200)
   If NOT (C1)
      OUTPUT ("value of a is not in range");
   END IF
   IF NOT (C2)
      OUTPUT ("value of b is not in range);
   END IF
   IF NOT (C3)
      OUTPUT ("value of c is not in range);
   END IF
UNTIL C1 AND C2 AND C3
OUTPUT ("The three sides of a Δle are", a, b, c);
(end of step1)
```

Step 2 : Is a Triangle

```
If (a < b+c) AND (b < a+c) AND (c < a+b)
    ~~out~~ Isatriangle = true
  else
        Is atriangle = false
[end q step 2]
```

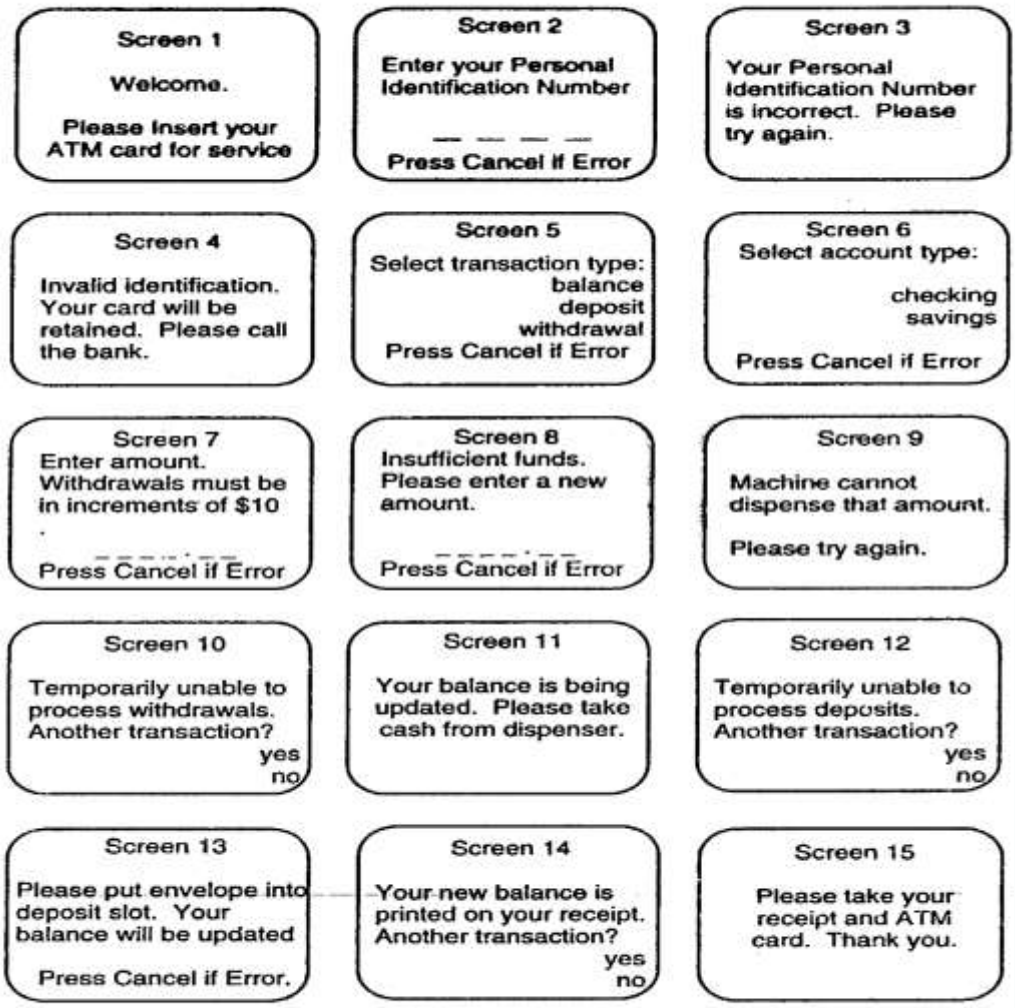
Step 3 : Determine triangle type

```
If
    Is atriangle
       if (a=b) AND (a=c)
            output (" equilateral triangle)
       else If (a!=b) AND (b!=c) AND (a!=c)
            output ("scalene triangle")
         else output ("Isosceles Δle")
    endif
     end if
  else output(not a triangle)
endif
(end q step 3)
```



c1: a<b+c?	F	T	T	T	T	T	T	T	T	T	T
c2: b<a+c?	—	F	T	T	T	T	T	T	T	T	T
c3: c<a+b?	—	—	F	T	T	T	T	T	T	T	T
c4: a = b?	—	—	—	T	T	T	T	F	F	F	F
c5: a = c?	—	—	—	T	T	F	F	T	T	F	F
c6: b = c?	—	—	—	T	F	T	F	T	F	T	F
a1: Not a triangle	X	X	X								
a2: Scalene											X
a3: Isosceles							X		X	X	
a4: Equilateral				X							
a5: Impossible					X	X		X			

5 (a) Explain SATM (Simple Automated Teller Machine) system. [10] CO1 L4



Screen 1
Welcome.
Please insert your ATM card for service

Screen 2
Enter your Personal Identification Number
— — — —
Press Cancel if Error

Screen 3
Your Personal Identification Number is incorrect. Please try again.

Screen 4
Invalid identification. Your card will be retained. Please call the bank.

Screen 5
Select transaction type:
balance
deposit
withdrawal
Press Cancel if Error

Screen 6
Select account type:
checking
savings
Press Cancel if Error

Screen 7
Enter amount. Withdrawals must be in increments of $10
— — — — . — —
Press Cancel if Error

Screen 8
Insufficient funds. Please enter a new amount.
— — — — . — —
Press Cancel if Error

Screen 9
Machine cannot dispense that amount.
Please try again.

Screen 10
Temporarily unable to process withdrawals. Another transaction?
yes
no

Screen 11
Your balance is being updated. Please take cash from dispenser.

Screen 12
Temporarily unable to process deposits. Another transaction?
yes
no

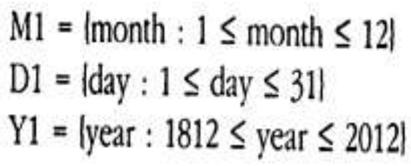
Screen 13
Please put envelope into deposit slot. Your balance will be updated
Press Cancel if Error.

Screen 14
Your new balance is printed on your receipt. Another transaction?
yes
no

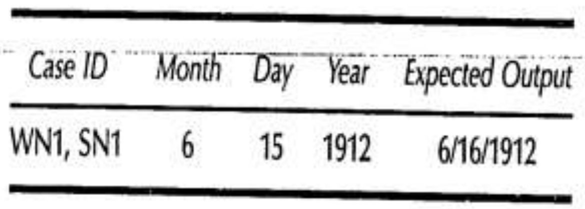
Screen 15
Please take your receipt and ATM card. Thank you.

6 (a) Perform Equivalence Class Testing on Next Date function and derive the test cases. [06] CO2 L3

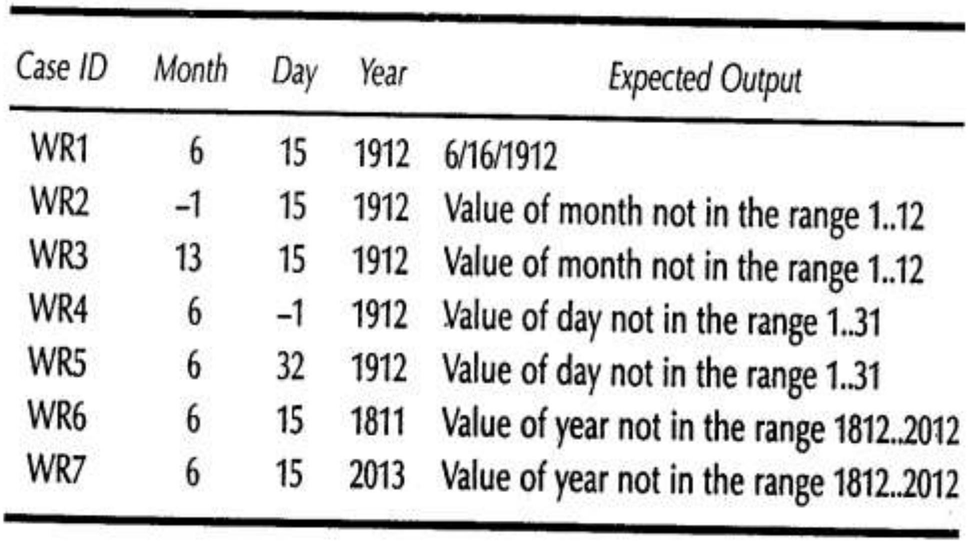
The valid range of values are



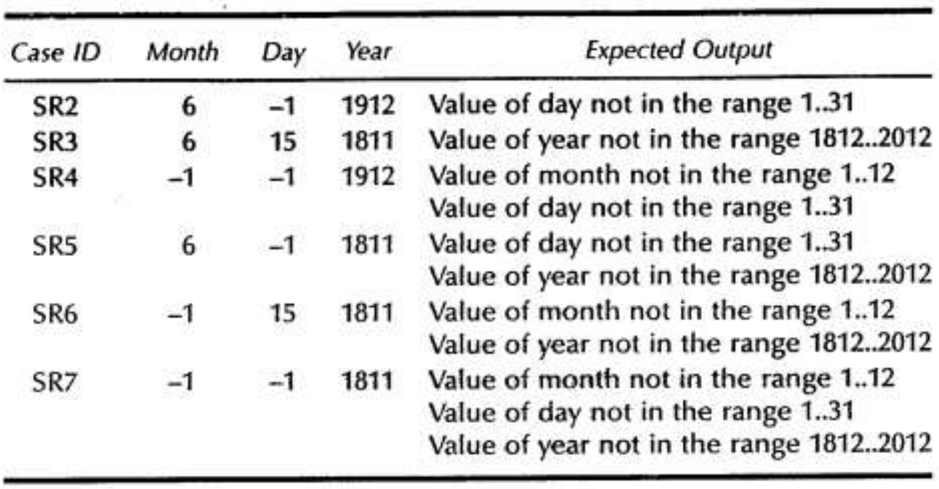
$$M1 = \{month : 1 \le month \le 12\}$$
$$D1 = \{day : 1 \le day \le 31\}$$
$$Y1 = \{year : 1812 \le year \le 2012\}$$



Case ID	Month	Day	Year	Expected Output
WN1, SN1	6	15	1912	6/16/1912



Case ID	Month	Day	Year	Expected Output
WR1	6	15	1912	6/16/1912
WR2	–1	15	1912	Value of month not in the range 1..12
WR3	13	15	1912	Value of month not in the range 1..12
WR4	6	–1	1912	Value of day not in the range 1..31
WR5	6	32	1912	Value of day not in the range 1..31
WR6	6	15	1811	Value of year not in the range 1812..2012
WR7	6	15	2013	Value of year not in the range 1812..2012



Case ID	Month	Day	Year	Expected Output
SR2	6	–1	1912	Value of day not in the range 1..31
SR3	6	15	1811	Value of year not in the range 1812..2012
SR4	–1	–1	1912	Value of month not in the range 1..12 Value of day not in the range 1..31
SR5	6	–1	1811	Value of day not in the range 1..31 Value of year not in the range 1812..2012
SR6	–1	15	1811	Value of month not in the range 1..12 Value of year not in the range 1812..2012
SR7	–1	–1	1811	Value of month not in the range 1..12 Value of day not in the range 1..31 Value of year not in the range 1812..2012

(b) Briefly explain the difference between

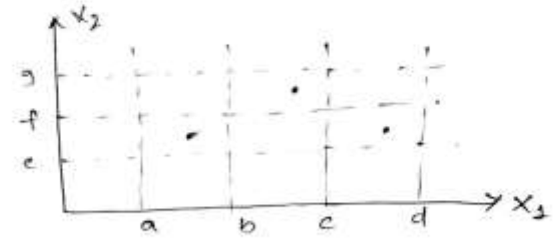
i) Weak Normal and Strong Normal Equivalence Class Testing.

ii) Weak Robust and Strong Robust Equivalence Class Testing.

[04] CO2 L4

(i)(a) weak normal Equivalence class testing

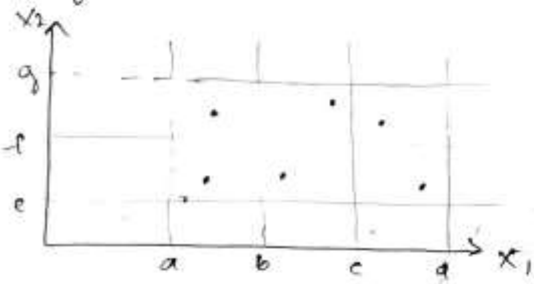
→ The test case variable value is taken one from each equivalence class



(b) strong Normal Equivalence class Testing

→ It is based on multiple fault

→ In this testcase values are from all the values from cardesion product of equivalince clas

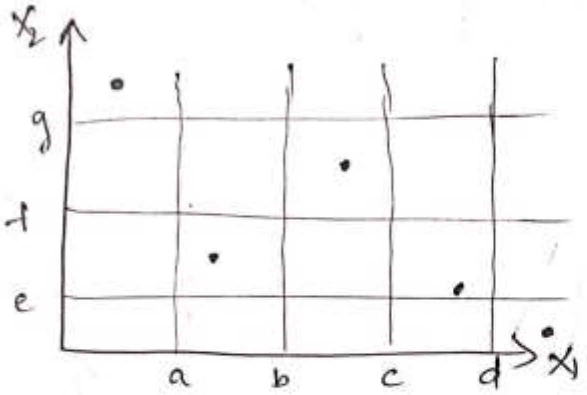


(ii) weak Robust ECT

→ It is the extension of weak normal equivalence class testing

→ If value is valid, each value is taken from each valid class

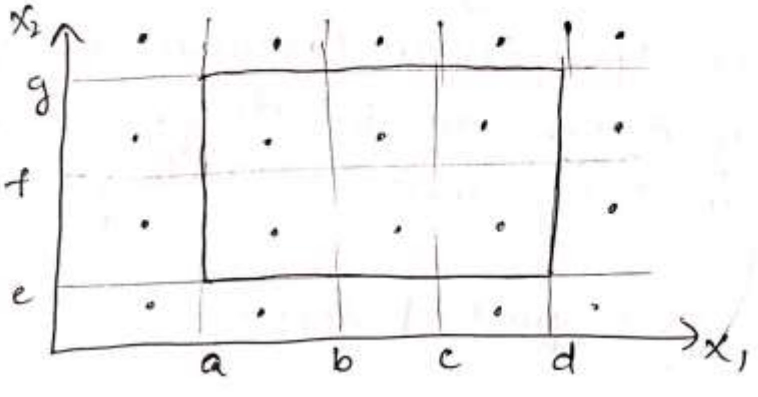
→ If value is invalid, one value must be invalid, and other variable values must be valid



(iv) Strong Robust ECT

→ It is the extension of strong normal equivalence clas testing

→ In this testcase values are from all the values from carstesion product of equivalence clase both valid and invalid


x_2
g
f
e
a
b
c
d
x_1