IT314
SOFTWARE ENGINEERING

Name: Nikhil J Jethanandani
ID: 202001108

SECTION A

Q1) Previous Date problem

Test Cases

Test Case ID	Day	Month	Year	Expected Output
1	1	6	2000	31-5-2000
2	2	6	2015	1-6-2015
3	2	6	2016	Invalid
4	1	1	1900	31-12-1899

5	31	12	1899	Invalid
6	31	12	1900	30-12-1900
7	29	2	2012	28-2-2012
8	1	3	2012	29-2-2012
9	29	2	2011	Invalid
10	30	2	2020	Invalid

Equivalence Class Partitions

Day:

Partition ID	Range	Status
E1	Between 1 and 28	Valid
E2	Less than 1	Invalid
E3	Greater than 31	Invalid
E4	Equals 30	Valid
E5	Equals 29	Valid for leap year
E6	Equals 31	Valid

Month:

Partition ID	Range	Status
E7	Between 1 and 12	Valid
E8	Less than 1	Invalid
E9	Greater than 12	Invalid

Year:

Partition ID	Range	Status
E10	Between 1900 and 2015	Valid
E11	Less than 1	Invalid
E12	Greater than 2015	Invalid

Q2)

Program 1

Equivalence Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{ 1,2,3,4,5,6,7,8};	7	6	6
2	{ 1,2,3,4,5,6,7,8};	9	-1	-1
3	{ 1,2,3,4,5,6,7,8};	A	Error	-1

Boundary Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{ 1,2,3,4,5,6,7,8};	8	7	7
2	{ 1,2,3,4,5,6,7,8};	1	0	0
3	{};	1	-1	-1

Program 2

Equivalence Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{ 1,2,3,2,4,5,2};	2	3	3
2	{ 1,3,5,7};	2	0	0
3	{2};	1	0	0
4	{1,2,3,4};	A	Error	0

Boundary Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{};	1	0	0
2	{ 1};	1	1	1

Program 3

Boundary Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{};	1	-1	-1

2	{ 1};	1	0	0
3	{2};	1	-1	-1

Equivalence Class Partitioning

Test Case ID	Arr Value	Target	Output	Expected Outcome
1	{ 1,3,5,7,9};	5	2	2
2	{ 1,3,5,7,9};	6	-1	-1
3	{1,2,3,4};	A	Error	0

Program 4

Boundary Class Partitioning

Test Case ID	a	b	c	Output	Expected Outcome
1	2	3	6	3	3
2	0	0	0	3	3
3	-1	2	2	3	3
4	999999999	999999999	9999999999	Error	3

Equivalence Class Partitioning

Test Case ID	a	b	c	Output	Expected Outcome
1	5	5	5	0	0
2	4	4	6	1	1
3	3	4	5	2	2

Program 5

Boundary Class Partitioning

Test Case ID	String 1	String 2	Output	Expected Outcome
1	“”	“”	true	true
2	abc	def	false	false
3	“”	def	true	true

Equivalence Class Partitioning

Test Case ID	String 1	String 2	Output	Expected Outcome
1	hello	helloworld	true	true
2	test	test	true	true
3	abced	bd	false	false

Program 6

(a) All equivalent classes

Class ID	Class
E1	All sides are positive
E2	two of its sides are zero
E3	One of its sides are negative
E4	Sum of two sides is less than third side
E5	Any of the side/sides is negative

(b) Identify test cases to cover the identified equivalence classes. Also, explicitly mention which test case would cover which equivalence class.

Test Case ID	Class ID	Test Case
T1	E1	A = 1, B = 1, C = 1
T2	E1	A = 3, B = 4, C= 5

T3	E2	A = 0, B = 0, C = 1
T4	E3	A = 0, B = 1, C = 2
T5	E4	A = 1, B = 3, C = 8
T6	E5	A = -1, C = 1, D = 5

(c) For the boundary condition A + B > C case (scalene triangle), identify test cases to verify the boundary.

A = 1, B = 3, C = 2 (d) For the boundary condition A = C case (isosceles triangle), identify test cases to verify the boundary.

A = 3, B = 2, C = 3 (e) For the boundary condition A = B = C case (equilateral triangle), identify test cases to verify the boundary.

A = 30, B = 30, C = 30 (f) For the boundary condition A2 + B2 = C2 case (right-angle triangle), identify test cases to verify the boundary.

A = 6, B = 8, C = 10 (g) For the non-triangle case, identify test cases to explore the boundary. A = 20, B = 10, C = 5 (h) For non-positive input, identify test points. A = 0, B = 10, C = 0

A = 0, B = 0, C = 0

A = 0, B = -1, C = 10

SCREENSHOTS:
They have not been uploaded as they are present in the lab PC and thus, I would update my lab repo whenever the lab is being accessed.

SECTION B

Control Flow Graph

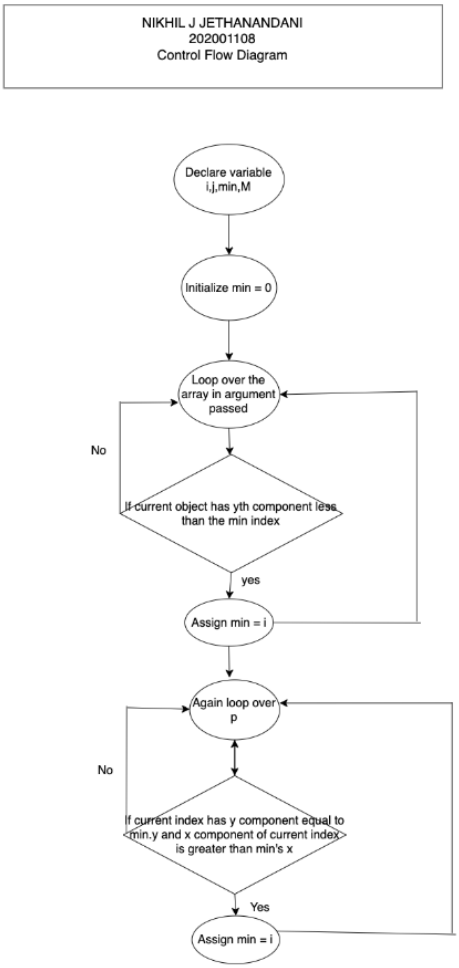

Diagram Link:

https://drive.google.com/file/d/1ju0I_bkEJBIrknq_tHbvTP5W0VuIDld5/view?usp=sharing

(2) Test Cases

(a) Statement coverage test set:In this all the statements in code should be covered

Test Number	Test Case
1	p is empty array
2	p has one point object
3	p has two points object with different y component
4	p has two points object with different x component
5	p has three or more point object with different y component

(b) Branch Coverage test set: In this all branch are taken at least once

Test Number	Test case
1	p is empty array
2	p has one point object
3	p has two points object with different y component
4	p has two points object with different x component
5	p has three or more point object with different y component

6	p has three or more point object with same y component
7	p has three or more point object with all same x component
8	p has three or more point object with all different x component
9	p has three or more point object with some same and some different x component

(c) Basic condition coverage test set:

Test Number	Test case
1	p is empty array
2	p has one point object
3	p has two points object with different y component
4	p has two points object with different x component
5	p has three or more point object with different y component
6	p has three or more point object with same y component
7	p has three or more point object with all same x component

8	p has three or more point object with all different x component
9	p has three or more point object with some same and some different x component
10	p has three or more point object with some same and some different y component
11	p has three or more point object with all different y component
12	p has three or more point object with all same y component

Each boolean expression has been evaluated to both true and false