Test Cases

Author- Avi Jain

1.)Test Case - 1

This case takes the input as Name (1-15 CHARACTERS) and Phone number (10 INTEGERS).
Testing Method used: *Boundary Value analysis*

Since there are 2 variables, Name and Phone number , the total number of test cases will be 4n+1; where n = 2
4*2+1 = 9

The set of boundary values is shown below:

Boundary Values	Name (1-15)	Phone number(10)
(Min) Value	A	3542781098
(min+) Value	Ad	4563890234
(max) Value	Chiraaag Guupta	5432678043
(max-) Value	Chaturvedi Sinha	4567890112
(Nom) Value	Pahwaaaa	9876545678

Using the values , Test cases can be designed as:

TEST CASE ID	Name	Phone number	Expected Output
1	(Min) A	0123466734	Valid- A,0123466734
2	(min+) AB	9876453724	Valid- AB,9876453724
3	(max) Ashwin Kumar Gu	9874583023	Valid- Ashwin Kumar Gu, 9874583023

4	(max-)Neermeet Pahwa		Invalid
5	(nom)Avi Jain	4567	Invalid
6	(Nom) Chaturvi	098765421	Invalid
7	(Nom) Yaadavna	4356287901564	Invalid
8	(Nom) Chiraagg	9842069420	Valid - Chiraagg, 9842069420
9	(Nom) Aadityaa	9733342555	Valid - Aadityaa, 9733342555

2.) Test case - 2

This case takes the input as the amount_of_waste(in kgs) ranging between 0-5 (integer)
Testing Method used: *Worst Case*

The set of boundary values is shown below:

Boundary Values	amount_of_waste(in kgs)
(Min) Value	0
(max) Value	5
(min+) Value	1
(max-) Value	4
(Nom) Value	3

Using the values , Test cases can be designed as:

Test Case ID	Amount Of Waste	Expected Output
1	0	Valid - 0 kgs
2	5	Valid - 5 kgs
3	6	Invalid
4	4	Valid - 4 kgs
5	3	Valid - 3 kgs

3.)Test Case - 3

Testing Technique used: ***Decision table testing***

Condition 1- With every Pickup, free 10 reward points will be given
Condition 2-If pickup is scheduled on weekly basis, no reward points will be given
Condition 3- With addition of every 10 levels , free 25 reward points will be given
Condition 4- With every pickup, 1 level will increase

The decision table for the program is shown below:

ENTRY-

		Rule 1	Rule 2	Rule 3
Condition Stub	C1: Each Pickup	T	F	0
	C2: Pickup Schedule = repeat weekly	0	F	0
	C3: Addition of every 10 levels	0	F	T
	C4; Each Pickup	0	T	F
Action Stub	A1: Free 10 Reward Points for scheduling daily	X		
	A2 : Free 25 Reward Points for scheduling daily			X
	A3: Increment of 1 level		X	

The test cases derived from the decision table are given below:

Test Case ID	No. of Pickup	Pickup Routine	No. of Levels	Incremented Levels	Expected Result
1	1	Daily	20	1	Free Reward Points
2	5	Weekly	30	1	Invalid Pickup Routine
3	2	Daily	21	1	Invalid Level Number
4	3	Daily	10	1	Free Reward Points
5	4	Daily	10	0	No level incremented

4.) Test Case - 4

Testing Technique used : *Decision Table testing*

Condition 1: Location is more than 10 km away from the centre ,Time taken will be more
Condition 2: Location is less than 10 km away from the centre ,Time taken will be less , then free reward points would be awarded to the customer

The decision table for the program is shown below:
ENTRY-

		Rule 1	Rule 2	Rule 3
Condition Stub	C1 : Location > 10 km & time taken is more	T	F	0
	C2 : Location < 10 km & time taken is less	F	T	0
Action Stub	A1 : Free reward points for Scheduling		X	

The test cases derived from the decision table are given below:

Test case ID	Location (kms)	Time Taken	Expected Result
1	11 km	less	Invalid Time Taken
2	6 km	less	Free Reward Points
3	5 km	more	Invalid Location (kms)

5.) Test case - 5

Testing Technique Used: *Boundary Value Analysis*
Case - Sign up components

Full Name - 1-15 characters
Email Address - use of character '@'
Permanent Address - 1-100 characters

Since there are three variables Full Name , Email Address , Permanent Address , the total number of test cases will be;
4n+1
Where n=3
4*3+1 = 12+1 = 13

The set of boundary values is shown below :

	Name (1-15)	Email Address (with @)	Permanent Address (1-100)
Min Value	1	@	1
Max Value	15	@	100
Min+ Value	2	@	2
Max- Value	14	@	99
Nom Value	8	@	50

Using the values , test cases can be designed as shown below :

Test Case ID	Full Name	Email Address	Permanent Address	Expected Output
1	A	a12@gmail.com	2-KarveNagar	Full Name - A Email Address - a12@gmail.com Permanent Address - 2-KarveNagar
2	Bk	bkbose420.in	D-34 kothrud	Invalid Email Address
3	Abhilasha	abhilashahumnain@gov.in	R-20 camp	Full Name - Abhilasha Email Address - abhilashahumnain@gov.in Permanent Address - R-20 camp
4	Abhinav_dy	abhinav.cool@sicsr.ac.in	S-26 sbroad	Full Name - Abhinav_dy Email Address - abhinav.cool@sicsr.ac.in Permanent Address - S-26 sbroad
5	Abhay	abhayoye@failure.com	g61 fcroad	Full Name - Abhay Email Address - abhayoye@failure.com Permanent Address - g61 fcroad
6	Anju	anuja.khale@cool.af	U-73 NIBM	Full Name - Anju Email Address - anuja.khale@cool.af Permanent Address - U-73 NIBM
7	Anjali	anjalilaila@lila.us	C-1 Rsparade	Full Name -Anjali Anjali Email Address - anjalilaila@lila.us anjalilaila@lila.us Permanent Address - C-1 RsparadeC-1 Rsparade
8	Ankur	ankurhurhur.com	Awas	Invalid Permanent Address and Email Address
9	Nimish	nimesh.jain@gmail.com	L	Full Name - Nimish Email Address - nimesh.jain@gmail.com Permanent Address - L
10	R	rboletor.ezpz	R2	Invalid Email Address and Full Name

11	Rohit	bhavnani@noob.com	Bavdhan 64east	Full Name -Rohit Email Address - bhavnani@noob.com Permanent Address - Bavdhan 64east
12	Watsin	shane33@icc.au	Sect2 VimanNgar	Full Name -Watsin Email Address -shane33@icc.au Permanent Address - Sect2 VimanNgar
13	Smith	bestplayereever@gmail.com	Star Mrket	Full Name - Smith Email Address -bestplayereever@gmail.com Permanent Address - Star Mrket

6.) Test case - 6

Testing Technique Used: ***Basic Path Testing--*** **Cyclomatic Complexity**

Case : Sign up Components

Username- A-Z , a-z (6 characters)

Password- in between 1-100 numbers

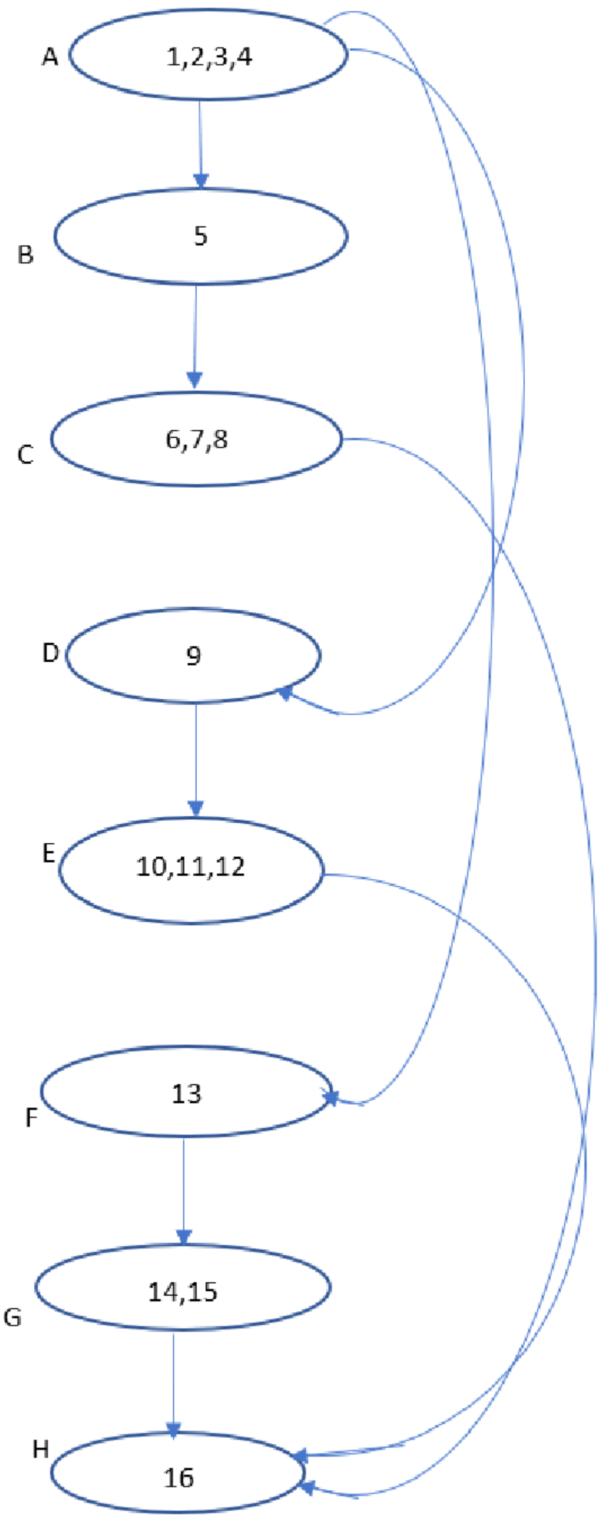
Consider the following program:-

```
main()
{
   int password;
```

```
    str username;
    printf("Enter the username");
    scanf("%s", &username);
    printf("Enter the password");
    scanf("%d", &password);

    if(password=="A" || password=="Z")
    {
        printf("Invalid password, please enter again");
    }
    else if (password < = 100)
    {
        printf("Valid password");
    }
    else{
        printf("Try again");
    }
} //end main
```

a.)Draw the DD graph for the above problem:



b) Cyclomatic Complexity :

(i) V (G) = e-n+2

Where; V(G) = cyclomatic number, e= no. of edges , n = no. of nodes

e=9 , n=8

V (G) = 9-8+2 = 1+2

V (G) = 3

(ii) cyclomatic no. of programs

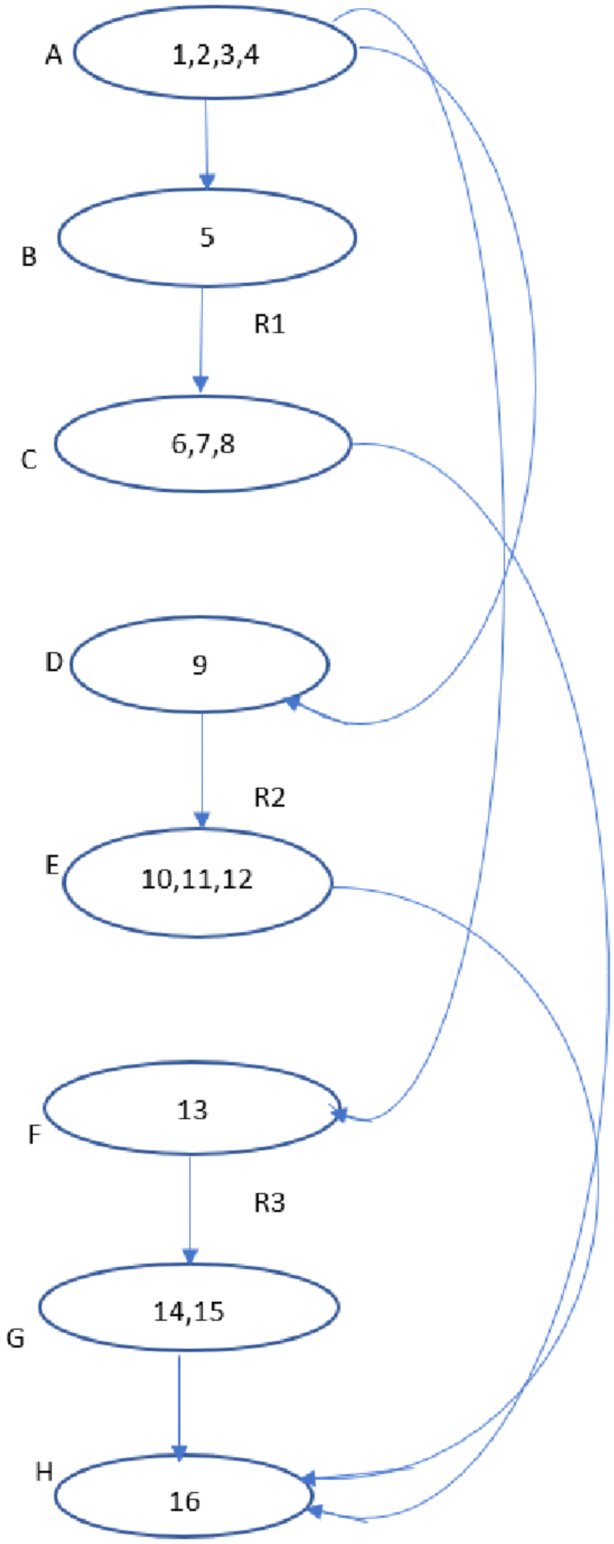
V (G) = d+1

Where; d=decision nodes

d=k-1 = 9-1 = 8

V (G) = d+1 = 8+1

V (G) = 9


A
1,2,3,4
B
5
R1
C
6,7,8
D
9
R2
E
10,11,12
F
13
R3
G
14,15
H
16

(iii) No. of Regions

V (G) = no. of regions

V (G) = 3 (R1,R2,R3)

c)Independent Paths :

Since the cyclomatic complexity of the graph is 3 , there will be 3 independent paths in the graphs as shown below -:

A-B-C-H

A-D-E-H

A-P-G-H

d)Test designs using the independent paths :

Test Case ID	Input Values	Expected Result	Independent paths covered by Test cases
1	A	Invalid password , please enter again	A-B-C-H
2	143	Valid password	A-D-E-H
3	019283746542	Try again	A-P-G-H

==