LAB EXPERIMENT – 1

Aim : Perform Adhoc Monkey testing.

Sol.

<u>Problem statement 1</u> : Write the test cases (use adhoc Monkey testing) to test a program which add two numbers, each number contain one/two digit(s).

Input : Input will be the two numbers, let's consider A & B.

A & B will be integers ranging from -99 <= A <= B <= 99.
Any other input other than these values, will be considered invalid.

Input domain : Input domain for the input values will be their type, i.e, **Integer**.

Output : Output will be the sum of the input values, i.e, **A + B.**

Output domain : Output domain for the output will be its type, i.e, **Integer**. If found anything else, it will be considered invalid.

No. of test cases required for this program :

There is no exact formula to calculate the no. of test cases required in a program. It could be infinite number of cases generated for a program, but here we will take only valid and some invalid test cases.

Test Cases :

T. NO.	INPUT		ACTUAL OUTPUT	VALID OUTPUT	RESULT
	A	B			
1.	-99	-99	-198	-198	VALID
2.	-100	1	-99	-	INVALID
3.	1	-1	0	0	VALID
4.	10	10	20	20	VALID
5.	"Hello"	"World"	"HelloWorld"	-	INVALID

00114807218
Ankit Gupta

6.	100	100	200	-	INVALID
7.	1	-99	-98	-98	VALID

How do you decide when you have 'tested enough' :

- 100% requirements coverage is achieved.
- More than 95% of test coverage and 100% functional coverage is achieved.
- Less than 5% Minor defect are open, and if open work around is available.
- All defects are retested and closed.
- All critical testcases are passed

<u>Problem statement 2</u> : Consider an automated banking application. The user can dial the bank from a personal computer, provide a six-digit password, and follow with a series of keyword commands thatactivate the banking function. The software for the application accepts data in the following form

Area Code Blank/ 3-digit number
Prefix 3-digit number not beginning with 0 or 1
Suffix 4-digit number
Password 6-character alphanumeric
Commands checkstatus, deposit, withdrawal
Design adhoc test cases to test the system.

Test Cases:

Input					**Expected Output**	**Actual Output**	**Result**
Area Code	**Prefix**	**Suffix**	**Password**	**Command**			
123	222	2323	sdf@es	checkstatus	Invalid	Invalid	Invalid
123	122	3232	3xfcsf	withdrawl	Invalid	Invalid	Invalid
	221	323	dsd2ds	deposit	Invalid	Invalid	Invalid

123	011	3332	Dfd2ss	checkstatus	Invalid	Invalid	Invalid
123	4567	2232	vg77rg	deposit	Invalid	Invalid	Invalid
123	523	43332	Sdsssf	withdrawal	Invalid	Invalid	Invalid
123	234	3456	32232	withdrawl	Invalid	Invalid	Invalid
123	23	3457	Efrf1d	withdrawal	Invalid	Invalid	Invalid
	234	12ab	Sdsssd	deposit	Invalid	Invalid	Invalid
012	345	2222	abc12fa	withdrawal	Invalid	Invalid	Invalid
97	345	2221	abde12	deposit	Invalid	Invalid	Invalid
321	657	7865	qwer34	withdrawal	*Accepted*	*Accepted*	*Accepted*
	345	6565	pswrd1	checkstatus	*Accepted*	*Accepted*	*Accepted*
123	234	3333	Dfew3E	deposit	*Accepted*	*Accepted*	*Accepted*
	203	3562	Asc4ls	deposit	*Accepted*	*Accepted*	*Accepted*