VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum – 590014, Karnataka State, India

A MINI PROJECT ON

"PYTHAGORAS THEOREM FOR RIGHT ANGLE TRIANGLE"

Submitted in partial fulfillment of assignment requirements for
Software Testing course and Laboratory

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

Submitted by:-

UDAY KUMAR J B **1SJ19IS118**

Under the guidance of:-

Prof. YOGARAJA G S R
Assistant Professor
Dept. of ISE, SJCIT
Carried out at
SOFTWARE TESTING (18IS62)
Course
SJCIT

S J C INSTITUTE OF TECHNOLOGY
Department Of Information Science And Engineering
Chickballapur – 562101
2021-2022

ACKNOWLEDGEMENT

With Great pride I would like to convey my heartfelt gratitude and appreciation to my alma-mater **“S.J.C Institute of Technology”** for giving me the required platform for the fulfillment of the project on Software Testing as per the V.T.U requirements, for the sixth semester.

I express my sincere thanks to **Dr. G.T.RAJU,** Principal of SJCIT for providing me with excellent infrastructure to complete the project. I express whole hearted gratitude to **Prof.SATHEESH CHANDRA REDDY** who is the respectable HOD of Information Science and Engineering Department. I wish to acknowledge his help in making my taskeasy by providing me with his valuable help and encouragement.

It is my pleasure to thank my guides **Prof. YOGARAJA G S R,** Assistant Professor, Department of Information Science and Engineering SJCIT for his guidance, encouragement and valuable suggestion from the beginning of the project work till the completion without which this project work would not have been accomplished. I am greatly indebted to him.

And last but not the least; I would be very pleased to express my heart full thanks to the teaching and non-teaching staff of the Department of Information Science and Engineering, SJCIT for their motivation and support. I also thank all those who extended their support and co-operation for successful completion of this project.

Uday Kumar J B	**1SJ19IS118**

ABSTRACT:

Testing is a technique that asses the performance of a software system application considering the different system configurations. This testing is done to determine the optimal configurations under which a system or an application can work fine without any bugs flaws in performance. So the most efficacious configuration that will deliver the required performance characteristics is spotted with the help of this testing. The second main reason for this testing is to verify the system's compatibility with the other software or equipment signified in the SRS (Software Requirement Specification).

There are two types of testing like manual testing and automated testing. In manual testing we have two types based on function and structure, they are:

Structural testing and functional testing they also called as white box and black box. In the structural testing which uses the internal design of the software for testing or in other words the software testing which is performed by the team which knows the development phase of the software, is known as structural testing. Structural testing tests basically tests different aspect of the software according to its types. Structural testing is just the opposite of behavioral testing.

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. If focuses on simulation of actual system usage but does not develop any system structure assumptions.

It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned about the source code of the software application. Each functionality of the software application is tested by providing appropriate test input , expecting interface the output and comparing the actual output with the expected output. This testing focuses on checking of user interface, APIs database, security, client or server application and functionality of the application under test.

Functional Testing Process:-

Functional testing involves the following steps:

1. Identify function that is to be performed.
2. Create input data based on the specifications of function.
3. Determine the output based on the specifications of function.
4. Execute the test case.
5. Compare the actual and expected output.

Content of report or Index:-

1. INTRODUCTION:-

The problem of Pythagoras theorem for right angled triangle has three inputs and can be tested by writing different test cases in a particular Range (1<inputs<50). Here we going to give the value of AB , BC and CA and we check that given triplet is a right angle or not.

$AB^2=BC^2+CA^2$

In this problem, we are taking only right angle triangle because Pythagoras theorem is only applicable for right angled triangle . For example AB, BC and CA Of a triangle are 5,4 and 3 the given is a right angled triangle else the test cases fail and finally conclude the program has some error . By using the input and range of a input we check the end result vs the expected output by preparing test cases for the given program.

Last but not List, In Software Testing we are doing testing on functional testing type this testing is done to increase the performance and to reduce the exception or errors, bugs, etc. We have a formula to find the number of test cases present in given range and the formula is 4b+1, where b be input and conditions . This is based on 3 types of analysis like Boundary value analysis, Decision table of the given Right angle triangle.

1.1CONDITIONS:-

Taking three inputs i.e AB,BC,CA from user . Based on the below mentioned condition the Pythagoras theorem helps in helps in forming a right angle triangle or not a Right angle triangle is given as follows:-

C1: AB!=BC

C2:BC!=CA

C3:CA!=AB

C4:$AB^2=BC^2+CA^2$

C5:AB>BC AB>C

2. PROBLEM STATEMENT:-

Design and develop a program in a language of your choice to solve the Right angle triangle problem define as follows:

Accept three integers which are supposed to be the three sides of a right angle triangle and determine if the three values represent an Right angle triangle, Not a triangle, Invalid input and out of range. Assume that the upper limit for the size of any side is 50 and lower is 1. Derive test cases for your program based on boundary-value analysis, Equivalence class testing, Decision table, execute the test cases and discuss the result of Right angle triangle.

Accept three inputs from the user like AB, BC and CA , find the production and status of Pythagoras theorem for right angle triangle according to the requirements, conduct boundary analysis to the program and test the code. Output like form a Right angle triangle or not a Right angle triangle, Invalid inputs and out of range. Assume that the upper limit for the size of any side is 50 and lower is 1.Derive test cases for your program based on boundary value analysis , Equivalence class testing , Decision table , execute the test cases and discuss the result of Right angle triangle.

3. REQUIREMENTS:-

R1. The system should accept 3 positive integer numbers (AB, BC, AC) which represents 3 sides of the Right angle triangle.

R2. Based on the input it should determine what is possibility or output of Right angle triangle.

R3. If the requirement R1 satisfied then the system should determine the type of output in Right angle triangle problem, can be

(I) Right angle triangle ($AB^2=BC^2+AC^2$)

(II) Not a Right angle triangle ($AB^2 \,!=BC^2+AC^2$)

(III)Invalid input (AB=BC or BC=CA or CA=AB)

(IIII)Out of range (1<=AB=BC=CA<=50)

R4. Upper Limit for the size of any sides is 50.

R5.Lower limit for the size of any side is 1.

4. DESIGN:

4.1 ALGORITHM:

```
Step1: Input AB , BC and CA three integer values which represents three sides of the Right angle triangle.
Step2: If (1<=AB<=50 and 1<=BC<=50 and 1<=CA<=50);
Step3: If ($AB^2=BC^2+AC^2$):Then
    Print Right angle triangle. Do step 6.
Step4: (AB=BC or BC=CA or CA=AB):
    Then    Print invalid input. Do step 6.
Step 5: else:
        print not a right angle triangle.
Step6: Stop. Print out of range.
```

5. PROGRAM CODE:

```
AB=int(input("Enter the value of side AB:"))
BC=int(input("Enter the value of side BC: "))
CA=int(input("Enter the value of side CA: "))
if 1<=AB<=50 and 1<=BC<=50 and 1<=CA<=50 :
  if (AB²==BC²+CA²):
        print ("Right angle triangle")
  elif (AB==BC or BC==CA or CA==AB):
        print ("Invalid input")
  else: print ("Not a right angle triangle")
else:
        print ("Out of range”)
```

6. TESTING:

6.1.1.TECHNIQUE USED: BOUNDARY VALUE

ANALYSIS 2.Test cases decision.

For BVA problem the test cases can be generation depends on the output. Here we least worried on the constraints on input domain.

The right angle triangle problem or Pythagoras theorem for triangle problem takes 3sides as input and check if for validity, hence n=3. Since BVA yields (4n+1) test cases according to single fault assumption theory , hence we say that total number of test cases will be (4*3+1)=13.

The maximum limit of each sides AB, BC and CA of the right angle triangle is 50 units according to requirements R4 and lower limit of 1unit.So AB , BC and CA

lies between

1<=AB<=50

1<=BC<=50

1<=CA<=50

BOUNDARY VALUE ANALYSIS:

A function F of 2 variables X1 and X2 .When implemented as a program will have same boundaries. A<=X1<=B,C<=X2<=D strongly typed languages permit explicit definition of such variables range. The boundary value analysis test cases are obtained by holding the values of all but are variable at their normal values.

Equivalence classes for AB:

E1: Value less than1.

E2: Value in the range.

E3: Value greater than 50.

Equivalence classes for BC:

E4: Value less than1.

E5: Value in the range.

E6: Value greater than50.

Equivalence classes for CA:

E7: Value less than1.

E8:Value in the range.

E9:Value greater than 50.

6.2.2EQUIVALENCE

CLASS :

They form a partition of a set where partition refers to a collection of mutually disjoint subset union of which is the entire set. This as two implication entire set. This as two implication of testing:

->Entire set represented provides a form completeness.

->Disjointedness ensures a form of non redundancy.

(II)TECHNIQUE USED: EQUIVALENCE CLASS PARTITION

(i)Test cases design

Equivalence class partitioning technique focus on the input domain, we can obtain a richer set of test cases. What are some possibilities for the three integers AB, BC and CA ? they canal be Right angle triangle, not Right angle triangle , Invalid input, out of range.

The maximum limit of each sides AB, BC and CA of the right angle triangle is 50 units according to requirement R4 and 1 units according to requirement R4 and 1 units to the requirement R5. So AB, BC and lies between

1<=AB<=50

1<=BC<=50

1<=CA<=50

6.3.3DECISION TABLE:

Decision table have been used to represent and analyse complex logical relationship.

4 Portions of decision table:

(I)The part to the left of the bold vertical line is the stub portion.

(II)To the right is entry portion.

(III)The part above the bold horizontal line is the condition.

(Iv) Portion and below is the action portion.

6.3.***3TECHNIQUE USED: DECISION TABLE APPROACH***

Decision table based testing has been around since he early 1960’s. It is used to depict complex logical relationships between input data. A decision table is method used to build a complete set of test cases without using the internal structure of the program in question. In order to create test cases we use a table to contain the input and output values of program.
The '_ _’ symbol in the table indicates don’t care values. The table shows the 5 conditions and 4 actions.

All the conditions in the decision table are binary value. Hence it is called as binary value . Hence it is called as Limited entry decision table.

7. TEST CASES TABLE:-

7.1 BOUNDARY VALUE TEST CASES:

TC	INPUTS			EXPECTED	ACTUAL	STATUS
	AB	BC	CA	OUTPUT	OUTPUT	
1	5	4	3	Right angle triangle	Right angle triangle	Pass
2	13	12	5	Right angle triangle	Right angle triangle	Pass
3	8	6	5	Not a Right angle triangle	Not a Right angle triangle	Pass
4	4	3	2	Right angle triangle	Not a Right angle triangle	Fail
5	25	24	7	Right angle triangle	Right angle triangle	Pass
6	17	15	8	Right angle triangle	Right angle triangle	Pass
7	9	8	7	Right angle triangle	Not a Right angle triangle	Fail
8	41	40	9	Right angle triangle	Right angle triangle	Pass
9	15	14	12	Right angle triangle	Not a Right angle triangle	Fail
10	30	20	10	Not a Right angle triangle	Not a Right angle triangle	Pass
11	12	10	10	Invalid input	Invalid input	Pass
12	8	8	8	Invalid input	Invalid input	Pass
13	10	10	12	Not a Right angle triangle	Invalid input	Fail

7.2 EQUIVALENCE CLASS TESTING:

7.2.1 WEAK NORMAL FORM:

TC	INPUTS AB	 BC	 CA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
WN1	5	4	3	Right angle triangle	Right angle triangle	Pass
WN2	8	6	5	Not a Right Angle triangle	Not a Right angle triangle	Pass
WN3	12	10	10	Invalid input	Invalid input	Pass
WN4	10	10	12	Invalid input	Invalid input	Pass
WN5	8	8	8	Invalid input	Invalid input	Pass

7.2.2 WEAK ROBUST EQUIVALENCE CLASS TESTING:

TC	INPUTS AB	 BC	 CA	EXPECTED OUTPUT	STATUS
WR1	-5	4	3	Value of AB not in the range	Pass
WR2	5	-4	3	Value of BC not in the range	Pass
WR3	5	4	-3	Value of CA not in the range	Pass
WR4	53	45	28	Value of AB not in the range	Pass
WR5	45	53	28	Value of BC not in the range	Pass
WR6	45	28	53	Value of AB not in the range	Pass

7.2.3 STRONG ROBUST EQUIVALENCE CLASS TEST CASES:

TC	INPUTS AB	BC	CA	EXPECTED OUTPUT	STATUS
SR1	-5	4	3	Value of AB not in the range	Pass
SR2	5	-4	3	Value of BC not in the range	Pass
SR3	5	4	-3	Value of CA not in the range	Pass
SR4	-5	-4	3	Value of AB and BC not in the range	Pass
SR5	5	-4	-3	Value of BC and CA not in the range	Pass
SR6	-5	4	-3	Value of AB and CA not in the range	Pass
SR7	-5	-4	-3	Value of AB,BC and CA not in the range	Pass

7.3 Decision table:-

TC ID	INPUTS			EXPECTED OUTPUT	STATUS
	AB	BC	CA		
DT 1	5	4	3	Right angle triangle	Pass
DT 2	6	5	4	Not a Right angle triangle	Pass
DT 3	9	8	7	Not a Right angle triangle	Pass
DT 4	8	8	8	Invalid input	Pass
DT 5	?	?	?	Impossible	Pass
DT 6	?	?	?	Impossible	Pass
DT 7	13	12	5	Right angle triangle	Pass
DT 8	?	?	?	Impossible	Pass
DT 9	12	10	10	Invalid input	Pass
DT 10	17	8	15	Right angle triangle	Pass
DT 11	10	10	12	Invalid input	Pass

8. Execution and Result Discussion:-

Execute the program against the designed test cases and complete the table for actual output columns and status column.

9. Test Report:-

1. Number of TC Executed:**42**

2. Number of Defects raised:**4**

3. Number of TC'S passed:**38**

4. Number of TC'S failed: **4**