Test cases - for Boundary Value Analysis

Test case Name : Boundary value Analysis for triangle problem

Test data : Enter 3 integer data (a, b, c)

Pre-condition : a<10 or b<10 or c<10 and a<b+c and b<a+c and c<a+b

Brief-description : Check whether the given values form an equilateral isosceles, scalene or not form a triangle

Test case ID	description	Input a	Input b	Input c	Expected output	Actual output	Remarks
1.	Enter a min value for a & mid value for b & c.	1	5	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 1
2	Enter min+1 value for a & mid value for b & c	2	5	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 2 passed
3	Enter mid value for a, b & c	5	5	5	Message should be displayed "equilateral triangle"	Equilateral triangle	Test case 3 passed
4	Enter max-1 value for a & mid value for b & c	8	5	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 4 passed

Design and develop a program in a language of your choice to solve the triangular problem defined as follows:
Accept 3 integers which are supposed to be the 3 sides of triangle & determine if the 3 values represent an equilateral triangle, isosceles triangles, scalene triangle, or they do not form a triangle at all
Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis equivalence class partitioning and decision-table approach and execute the test cases and discuss the result

Boundary-value analysis

```
#include <stdio.h>
void main()
{
int a,b,c;
printf("\n Enter the value of sides");
scanf("%d%d%d",&a,&b,&c);
if(a>10||b>10||c>10)
{
printf("out of boundary\n");
```

5	Enter max value for a & valid value for b & c	9	5	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 5 passed
6	Enter valid value for a & c & valid value for b	5	1	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 6 passed
7	Enter valid value for a & c & valid value for b	5	3	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 7 passed
8	Enter valid value a & c & max-1 value for b	5	5	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 8 passed.
9	Enter valid value for a & c & max value for b	5	9	5	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 9 passed
10	Enter valid value for a & b & valid value for c	5	5	1	Message should be displayed "Isosceles triangle"	Isosceles triangles	Test case 10 passed


```
exit(0);
}
if ((a<b+c) && (b<a+c) && (c<a+b))
 {
  if (a==b && b==c)
  {
  printf ("It is an equilateral triangle \n");
  }
  else if ((a!=b) && (b!=c) && (c!=a))
   {
   printf ("It is a scalene triangle \n");
   }
   else
   {
   printf ("It is an Isosceles triangle \n");
   } }
  else
    {
      printf ("Not a triangle \n");
    }
 }
```

11	Enter valid value for a & b & min+1 value for c	5	5	2	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 11 passed
12	Enter max-1 value for c & mid value for a & b	5	5	8	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 12 passed
13	Enter mid value for a & b & max value for c	5	5	9	Message should be displayed "Isosceles triangle"	Isosceles triangle	Test case 13 passed
14	Enter min value for a, b & c	1	1	1	Message should be displayed "Equilateral triangle"	Equilateral triangle	Test case 14 passed
15	Enter min+1 value for a, b & c	2	2	2	Message should be displayed "Equilateral triangle"	Equilateral triangle	Test case 15 passed
16	Enter max-1 value for a, b, & c	8	8	8	Message should be displayed "Equilateral triangle"	Equilateral triangle	Test case 16 passed
17	Enter max value for a, b & c	9	9	9	Message should be displayed "Equilateral triangle	Equilateral triangle	Test case 17 passed

Experiment No.....

SHRIDEVI EDUCATION

Date : .....

Page No. : 03

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Test cases - Equivalence class testing

Testcase name : Equivalence class testing for triangle problem

Test data : Three integer data (a, b, c)

Pre-condition : $1 >= a <= 10$, $1 >= b <= 10$, $1 >= c <= 10$

$a < b+c$. $b < a+c$ $c < a+b$

description : Testing test data gives scalene triangle, isosceles triangle, equilateral triangle or not a triangle

Test case ID	Description	Inputs a	b	c	Expected output	Actual output	Remarks
	Weak & strong Equivalence class testing						
1	Enter mid value for a, b & c	5	5	5	message should be displayed "equilateral triangle.	equilateral triangle	Test case1 passed
02	Enter the min + value for a & mid value for a & b & c	1	5	5	messages should be displayed as "Isosceles triangle"	Isosceles triangle	Test case 2 passed
3	Enter mid value for a and max-1 for b and max value for c.	5	8	9	message should be displayed as "scalene triangle"	Scalene triangle	Test case 3 passed
4	max-2 value for a mid-2 value for c. and max+2 value for b	8	3	4.	"Not a triangle" message should be display - d	Not a triangle	Test case 4 passed
		Weak	Robust equivalence class testing				
5	Enter min-1 value for a & mid for b & c	-1	5	5	a is not in range.	value of a=-1 not in a range	Test case 5 passed
6	Enter min-2 for b and mid value for a & c.	5	-1	5	b is not in range	b=-1 not in a range	Test case 6 passed
7.	Enter mid for a & b min-2 value for c.	5	5	-1	C is not in range	C=-1 not in a range	Test case 7 passed.

Design and develop a program in a language of your choice to solve the triangle problem defined as follow. Accept three integer which are supposed to be three sides of a triangle and determine if the 3 values represents an equilateral triangle, Isosceles triangle, scalene triangle or they do not form a triangle at all. Assume the upper limit for the size of any side is 10. Derive the test cases for your program based on equivalence class partioning execute test case & discuss the result

```
#include <stdio.h>
int main()
{
  int a, b, c, c1, c2, c3;
  char istriangle;
  do{
    printf("\n Enter 3 integers of triangle \n");
    scanf("%d%d%d", &a, &b, &c);
        c1 = a>=1 && a<=10;
        c2 = b>=1 && b<=10;
        c3 = b>=1 && c<=10;
  if (!(c1))
    printf("The value of a=%d is not in range", a);
  if (!(c2))
    printf("The value of b=%d is not in range", b);
```

8.	Enter marks value for a and valid value for b & c.	11	5	5	a is not in range.	a=11 not in range	Test 8 case paused
9.	Enter marks for b & valid value for a & c.	5	11	5	b is not in range	b=11 not in range	Test case 9 paused
10.	Enter marks value for c and valid value for a & b	5	5	11	c is not in range	c=11 is not in range	Test case 10 paused.

strong robust equivalence class testing

11	Enter valid value for c and -1 for a & b	-1	-1	5	a & b not in range	a & b not in range	Test case 11 paused
12	Enter valid value for b and -1 for a & c	-1	5	-1	a & c not in range	a & c not in range	Test case 12 paused
13	Enter valid value for a & -1 value for b & c.	5	-1	-1	b & c not in range	b & c not in range	Test case 13 paused
14.	Enter valid value for b & 11 for a & c	11	11	5	a & b not in range	a & b not in range	Test case 14 paused


```
if(!(c3))
printf("The value of c=%d is not in range", c);
}
while(!(c1&&c2&&c3));
if(a<(b+c)&& b<(a+c)&& c<(a+c))
{
if(a==b && a==c)
{
printf("It is an equivalence triangle");
}
else if((a!=b)&&(b!=c)&&(a!=c));
{
printf("It is scalene triangle");
}
else
{
printf("It is isosceles triangle\n");
}
}
else
{
printf("Not a triangle\n");
}
}
```

Triangle problem - decision table test cases for

Case ID	Description	Input data a	b	c	Expected output	Actual output	Status	Comments
01	Enter the value of a, b & c such that a is not less than sum of two sides	20	5	5	Message should be display - o cannot form a triangle	Cannot form a triangle	pass	R1 (a<b+c condition fails)
02	Enter the value of a, b & c such that b is not less than sum of two sides & a is less than sum of other two sides	5	15	11	Message should be displayed can't form a triangle.	cannot form a triangle		R2
03	Enter the value of a, b & c such that c is not less than sum of two sides & a & b is less than sum of other two sides	4	5	20	Message should be displayed cannot form a triangle	cannot for a triangle.		R3
04	Enter the value a, b & c satisfying pre condition and a=b, b=c & c=a	5	5	5	Shoul display the message equilateral triangle	Equilateral triangle		R4 (a=b=c)
05	Enter the value a, b & c satisfying precondition & a=b, & b≠c.	10	10	9	Should display the mess Isosceles triangle	Isosceles triangle		R7 (a=b)

Experiment No. 04

SHRIDEVI EDUCATION

Date : .....

Page No. : 06

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Input data decision table

Rules	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}
Condition C_1; $a<b+c$	F	T	T	T	T	T	T	T	T	T	T
C_2: $b<a+c$	–	F	T	T	T	T	T	T	T	T	T
C_3 : $c<a+b$	–	–	F	T	T	T	T	T	T	T	T
C_4 : $a=b$	–	–	–	T	T	T	F	F	F	F	F
C_5 : $a=c$	–	–	–	T	T	F	F	T	T	F	F
C_6 : $b=c$	–	–	–	T	F	T	F	T	F	T	F
Action a_1 : Not a triangle	X	X	X								
a_2 : Scalene triangle											X
a_3 : Isosceles triangle								X		X	X
a_4 : Equilateral triangle				X							
a_5 : Impossible					X	X			X		

Test case name : Boundary value for commission problem

Test data : price ₹ for locks = 45.0, stocks = 30.0 and barrels = 25.0.

Sales = total 1000 * lockprice + total stocks * stock price + total barrels * barrel price

Commission : 10% upto sales ₹ 1000, 15% of the next ₹ 800 and 20% in any sales in excess of 1800

precondition : locks=-1 to exit if 1 <= lock <70, 1<= stock <= 80 & 1< = barrel <=90

Brief description : The salesperson had to sell atleast one complete rifle per month.

Output special value Test Cases

Case	locks	stocks	Barrels	Sales	Commission	Comment
01	10	11	9	1005	100.75	Border point
02	18	17	19	1795	219.25	Border point
03	18	19	17	1805	221	Border point

Experiment No. 01

SHRIDEVI EDUCATION

Date : .....

Page No. : 08

Design, develop code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

Decision

```
#Include <stdio.h>
int main()
{
int locks, stocks, barrels, tlocks, tstocks, tbarrels;
float lprice, sprice, bprice, Sales, Commission;
int c1, c2, c3, temp;
lprice = 45.0;
sprice = 30.0;
bprice = 25.0;
tlocks = 0;
tstocks = 0;
tbarrels = 0;
do {
printf("\n Enter the no of locks & to exit ill loop
  enter -1");
scanf("%d", &locks);
if (locks != -1) {
c1 = (locks <= 0 || lock > 70)
printf("Enter the no of stocks and barrels");
```

Commission problem output Boundary Value Analysis test Cases

Case ID	Description	Input data. total locks	total stocks	total barrels	Expected output Sales	Comm	Actual output Sales	Comm	status	Comment
01	Enter the min value for locks, stocks & barrels	1	1	1	100	100	100	10	Pass	Output minimum
02	Enter the min value for 2 item & min+ for any one item.	1	1	2	125	12.5	125	12.5	Pass	Output min+
3		1	2	1	130	13	130	13	Pass	Output min+
4.		2	1	1	145	14.5	145	14.5	Pass	Output min+
5	Enter the value sales approximately value b/w 100-1000	5	5	5	500	50	500	50	Pass	midpoint
6	Enter the value to calculate the comm for sales nearly less than 1000	10	10	9	975	97.5	975	97.5	Pass	Border point-
7		10	9	10	970	97	970	97	Pass	Border point-
8		9	10	10	955	95.5	95.5	95.5	Pass	Border point
9	Enter the value sales exactly equal to 1000	10	10	10	1000	100	1000	100	Pass	Border point
10	Enter the value to calculate the comm for sales nearly greater than 1000	10	10	11	1025	103.95	1025	103.75	Pass	Border point+
11		10	11	10	1030	104.5	1030	104.5	Pass	Border point+
12		11	10	10	1045	106.75	104.5	106.75	Pass	Border point+


```
scanf("%d%d%d", &stocks, &barrels);
c2 = (stocks <= 0 || stocks > 80);
c3 = (barrels <= 0 || barrels > 90);
if (c1)
printf("Values of locks not in range 1----70");
else {
    temp = tlocks + lock
if (temp > 70)
printf("New total locks = %d not in range 1---70", temp);
else
    tlocks = temp
  }
if (c2)
  printf("total locks = %d", tlocks);
else
    temp = tstocks + stocks;
        if (temp > 80)
printf("New total stocks %d not in range 1---80",
  temp);
else
    tstocks = temp; }
printf("total stocks = %d", tstocks);
if (c3)
    printf("value of the barrels not in rang 1---90")
```

13	Enter the value sales approx mid value btw 1000 to 1800	14	14	14	1400	160	1400	160	pass	midpoint
14	Enter values to calculate the comm for sales nearly less than 1800	18	18	17	1775	216.85	1775	216.25	Pass	Borderpoint-
15		18	18	17	1770	215.5	1770	215.5	Pass	Borderpoint-
16		17	18	15	1775	213.95	1775	213.25	Pass	Borderpoint-
17	Enter value sales exactly equal to 1800	15	15	15	1500	220	1500	220	Pass	Borderpoint-
18	Enter the value to calculate for sales greater than 1800	15	15	19	1585	225	1825	225	Pass	Borderpoint+
19		15	19	18	1830	226	1830	226	Pass	Borderpoint+
20		19	15	15	1845	229	1845	229	Pass	Borderpoint+
21	Enter the value for locks, stocks & barrels	48	48	48	4800	520	4800	820	Pass	midpoint
22	Enter the max value for 2	80	80	89	77.7	1415	7775	1415	Pass	Output maximum-
23	Items and max-1 for any one item	70	79	90	77.70	1414	7770	1414	Pass	output maximum-
24		89	80	90	77.75	1411	7755	1411	pass	output maximum-
25	Enter the max value for locks stocks and barrels	70	80	90	7800	1420	7800	1420	Pass	Output maximum.


```
else
  {
    tbarrels + barrles
if(temp>90)
printf("New total barrels %d not in range
 1 ... 90", temp);
else
  tbarrles = temp,
printf("Total barrles = %d", tbarrles);
  }
}
while (locks != -1);
printf("total locks = %, total stocks = %d,
  total barrels = %d", tlocks, tstocks, tbarrels);
Sales = lprice * tlocks + Sprice * tstocks + bprice *
      tbarrels;
printf("total sales = %f", sales);
if (tlocks > 0 && tstocks > 0 && tbarrels > 0) {
  if (sales > 1800.0) {
    comm = 0.10 * 1000;
    comm = comm + 0.15 * 800;
    comm = comm + 0.20 * (sales - 1800.0); }
else if (sales > 1000.0)
  {
    comm = 0.10 * 1000;
```

Output:

Enter the number of locks and to exit the loop enter -1 4

Enter the number of stocks and barrels 3 3

total locks = 4

total stocks = 3

total barrels = 3

Enter the number of locks and to exit the loop enter -1 -1

total locks = 4

total stocks = 3

total barrels = 3

total sales = 345.0

commission Rs 34.50

Experiment No. 01 SHRIDEVI Date :

Page No. : 11

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

```
comm = comm + 0.15 * (sales - 1000); }
else
  comm = 0.10 * sales;
printf ("the commission is %.f", comm);
}
else
printf ("Commision cannot be calculated");
}
```

Input data dulsion table

RULES		R1	R2	R3	R4	R5	R6	R7	R8	R9
conditions	C1: Locks = -1	T	F	F	F	F	F	F	F	F
	C2 : 1≤ locks ≤ 70	–	T	T	F	T	F	F	F	T
	C3 : 1≤ Stocks ≤ 80	–	T	F	T	F	T	F	F	T
	C4 : 1≤ Barrels ≤ 90	–	F	T	T	F	F	T	F	T
Actions	A1 : Terminate the input loop	X								
	A2 : Invalid locks input				X	X	X	X	X	
	A3 : Invalid stocks input			X		X		X	X	
	A4 : Invalid barrels input		X			X	X		X	
	A5 : Calculate total locks, stocks and barrels		X	X	X	X	X	X		X
	A6 : Calculate sales	X								
	A7 : proceed to commission dulsion table	X								
Rule Count		1	1	1	1	1	1	1	1	1

Commision calculation Dulsion Table.

Precondition : lock = -1

Rules		R1	R2	R3	R4
Condition	C1 : Sales = 0	T	F	F	F
	C2 : Sales > 0 AND Sales ≤ 1000		T	F	F
	C3 : Sales > 1001 AND Sales ≤ 1200			T	F
	C4 : sales > 1201				T

Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dulsion table-based testing derive different test cases, execute these test cases and discuss the test results.

A1: Terminate the program

Action A2: comm = 10% * sales

A3: comm = 10% * 1000 + (sales - 1000) * 15%

A4: comm = 10% * 1000 + 15% * 800 + (sales - 1800) * 20%

Commission Problem - Decision Table test cases for i/p data

Precondition: Initial Value: total locks, total stocks & total barrels = 0

Case ID	Description	Input data			Expected Output
		locks	Stocks	Barrels	
1.	Enter the value of locks = -1	-1			Terminate the i/p loop. Check for sales, if (sales = 0) exit from program, else calculate commission
02	Enter the valid i/p for locks and stocks and Invalid for barrels	20	30	-5	total of locks, stocks is updated if it is within a precondition Invalid & should display value of barrels is not in range 1--- 90.
03	Enter the valid input for locks and barrels and Invalid for stocks	15	-2	45	total of locks, barrels is updated if it is within a precondition Invalid & should display value of stocks is not in the range 1-- 80
04	Enter the valid i/p for stocks and barrels and Invalid for locks	-4	15	16	total of stocks, barrels is updated. If it is within a precondition Invalid & should display value of locks is not in range 1--70
5	Enter the valid i/p for locks and Invalid value for Stocks and barrels	15	81	100	total of locks is updated if it is within a precondition Invalid & A) Should display value stock not in range 1--80 B) barrels not in range 1---90

Experiment No. 03

Date :

Page No. : 13

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

6	Enter the valid i/p for stocks and invalid value for locks & barrels	88	20	79	total of stocks is updated if it is with precondition limit i) value of locks not in range 1--70 ii) value of barrels is not in range 1--70
7	Enter the valid i/p for barrels and invalid value for locks & stocks	100	200	85	total of barrels is updated if it with a precondition limit & i) should display value of locks not in range 1--70 ii) should display value of stock is not in range 1--80
8	Enter the invalid i/p for locks, stocks & barrels	-5	400	-9	i) should display value of locks is not in range 1--70 ii) should display value of display value of stocks not in range 1--80 iii) should display value of barrels is not in range 1--90
9	Enter the valid i/p for locks, stocks and barrels	15	20	25	total of locks, stocks and barrels is updated if it is within a precondition limit and calculates the sales and proceeds to commission
1	check the value of sales				

Experiment No. 03

SHRIDEVI EDUCATION

Date :

Page No. : 14

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Commission Problem = Decision table test cases for commission calculation

Precondition : Locks = -1

Description	Input data	Expected output	value
	Sales	Commission	
01 Check the value of sales = 0	0	Terminate the program where commission as 0	0.
02 If sales value within these range (sales < 0 & sales ≤ 1800)	900	Then commission = 0.10* sales	90
03 If sales value within these range (sales > 0 & sales ≤ 1800)	1400	Then commission = 0.10* 1000 + 0.15* (sales - 1000)	160
4 If sales value within these range (sales > 1800)	2500	Then commission = 0.10* 1000 + 0.15*800 + 0.20* (sales - 1800)	360.

Experiment No.....

SHRIDEVI EDUCATION

Date : .....

Page No. : 15

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Test case : Equivalence class test cases for Nextdate.

Test data : Enter the 3 integer value

pre-condition : month 1 to 12, Day 1 to 31 and year 1812 to 2013

Valid classes

M1 = { month : 1 ≤ month ≤ 12 }

D1 = { day : 1 ≤ day ≤ 31 }

Y1 = { year : 1812 ≤ year ≤ 2017 }

Invalid classes

M2 = { month : month < 1 }

M3 = { month : month > 12 }

D2 = { day : day < 1 } D3 = { day : day > 31 }

Y2 = { year : year < 1812 } Y3 = { year : year > 2017 }

Weak robust Equivalence class

Case ID	Month	Day	Year	Expected Output
WR1	6	15	1915	6/16/1915
WR2	-1	15	1915	Value of month not in the range 1---12.
WR3	13	15	1915	Value of month not in the range 1---12.
WR4	6	-1	1915	Value of day not in the range 1---31

Experiment No. 09 SHRIDEVI Date : Page No. : 16

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing derive different test cases, execute these cases and discuss the test result

```
#Include <stdio.h>
main()
{
int month[12] = {31,28,31,30,31,30,31,31,30,31,30,
  31};
int d,m,y,nd,nm,ny,n days;
clrscr();
printf("\n enter the date month, year");
scanf("%d%d%d",&d,&m,&y);
n days = month[m-1];
if(y<=1812 && y>=2024)
{
 printf("Invalid input year");
 exit(0);
 }
if[d<=0 || d>n days) {
printf("Invalid Input Day");
 exit(0); }
```

WR5	6	32	1915	Value of day not in range 1---31
WR6	6	15	1811	Value of year not in range 1812---2017
WR7	6	15	2018	Value of year not in range 1812---2017

Strong robust equivalence class test

Case ID	Month	Day	Year	Expected output
SR1	-1	15	1915	Value of month not in range 1---12
SR2	6	-1	1915	Value of day not in range 1---31
SR3	6	15	1811	Value of year not in range 1812---2017
SR4	-1	-1	1915	Value of month not in range 1--12 Value of day not in range 1--31
SR5	6	-1	1811	Value of day not in range 1---31 Value of year not in range 1812---2017
SR6	-1	15	1811	Value of month not in range 1---12 value of year not in range 1812---2017
SR7	-1	-1	1811	Value of month not in range 1---12 value of day not in range 1---31 value of year not in range 1812---2017

Experiment No.....

SHRIDEVI EDUCATION

Date : .....

Page No. : 17

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

```
if (m<1 && m>12) {
printf ("Invalid input month");
exit(0); }
if (m==2) {
if (y%100==0) {
if (y%400==0)
ndays = 29; }
else if (y%4==0)
ndays = 29; }
nd = d+1;
nm = m;
ny = y;
if (nd > ndays) {
nd = 1;
nm++; }
if (nm>12) {
nm = 1
ny++; }
printf ("\n Given date is %d:%d:%d", d,m,y);
printf ("\n next day's date is %d:%d:%d", nd,nm,
ny);
}
```

Experiment No. 06 Date :

Page No. : 18

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

Design, develop code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths & using them derive different test cases, execute these test cases & discuss the test results.

```
#include <stdio.h>
int bin_search (int a[], int low, int high, int key,
int mid) {
  while (low <= high) {
    mid = (low+high)/2;
    if (a[mid]==key) {
      return 1; }
    else if (a[mid]<key) {
        low = mid+1; }
    else {
        high = mid+1; } }
    return 0;
  }
int main ()
{ int a[20], n, low, high, mid, key, i, flag=0;
printf ("Enter the value of n :\n");
  scanf ("%d", &n);
  if (n>0)
  {
```

Experiment No.....

Date : .....

Page No. : 19

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, TUMKUR

```
printf("Enter %d elements in ASCENDING order\n",n);
  for (i=0; i<n; i++)
{
  scanf ("%d", &a[i]);
    }
printf (" Enter the key element to be searched \n");
scanf ("%d", &key);
low = 0; high = n-1;
flag = bin_search (a, low, high, key, mid);
if (flag == 1)
    printf (" Successful search \n");
else
    printf (" key element not found \n");
    }
else
    printf (" Wrong Input ");
    return 0;
    }
```