

## EXPERIMENT 2

I. Write a program to generate test cases using Boundary Value approach.

### SOURCE CODE

```
#include <stdio.h>
#include <conio.h>
int findmaxvalue(int *r, int s) {
    int max = 0, i;
    for (i = 0; i < s; i++) {
        if (max < r[i])
            max = r[i];
    }
    return max;
}
int main() {
    int n, i, j, k, id = 0, min[3], max[3];
    int ub[20], up[3], var[3], t_id = 0;
    int lb[20], row[20], dow[20];
    printf("\nEnter no. of variables: ");
    scanf("%d", &n);
    printf("\nEnter max and min limit for variables: \n");
    for (i = 0; i < n; i++) {
        printf("\n min[%d]:", i + 1);
        scanf("%d", &lb[i]);
        printf("\n max[%d]:", i + 1);
        scanf("%d", &ub[i]);
    }
    for (i = -1; i < n; i++) {
        if (i == -1)
            printf("\nT_id\t");
        else
            printf("var%d\t", i + 1);
    }
    printf("Expected Output\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < 4; j++) {
            printf("%d", ++t_id);
            for (k = 0; k < n; k++) {
                if (k == i) {
                    if (j == 0) {
                        row[k] = lb[i];
                        printf("\t%d", lb[i]);
                    }
                    else if (j == 1) {
                        row[k] = lb[i] + 1;
                        printf("\t%d", lb[i] + 1);
                    }
                }
            }
        }
    }
}
```

```

    }
    else if (j==2) {
        row[k] = ub[i]-1;
        printf("\t%d",ub[i]-1);
    }
    else {
        row[k] = ub[i];
        printf("\t%d",ub[i]);
    }
}
else {
    row[k] = (ub[k]+lb[k])/2;
    printf("\t%d", (ub[k]+lb[k])/2);
}
}
printf("\t%d\n", findmaxvalue(row,n));
}
}
printf("%d", ++t_id);
for (i=0; i<n; i++) {
    row[i] = (ub[i]+lb[i])/2;
    printf("\t%d", (ub[i]+lb[i])/2);
}
printf("\t%d\n", findmaxvalue(row,n));
getch();
return(0);
}

```

**OUTPUT:**

```

D:\College\ST Lab\auto_boundary.exe
Enter no. of variables: 5
Enter max and min limit for variables:
min[1]:1
max[1]:100
min[2]:1
max[2]:100
min[3]:1
max[3]:100
min[4]:1
max[4]:100
min[5]:1
max[5]:100
t_id  var1  var2  var3  var4  var5  Expected Output
1      1      50    50    50    50    50
2      2      50    50    50    50    50
3      99     50    50    50    50    99
4      100    50    50    50    50    100
5      50     1     50    50    50    50
6      50     2     50    50    50    50
7      50    99     50    50    50    99
8      50   100     50    50    50    100
9      50     1     50    50    50    50
10     50     2     50    50    50    50
11     50     50    99     50    50    99
12     50     50   100     50    50    100
13     50     50     50     1     50    50
14     50     50     50     2     50    50
15     50     50     50    99     50    99
16     50     50     50   100     50    100
17     50     50     50     1     50    50
18     50     50     50     2     50    50
19     50     50     50     50    99    99
20     50     50     50     50   100    100
21     50     50     50     50    50    50

```