

Program 1,2,3)

Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, 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, execute the test cases and discuss the results.

```
#include<stdio.h>
#include<conio.h>
int main( )
{
int a,b,c,c1,c2,c3; do
{
printf("enter the sides of triangle\n");
scanf("%d%d%d",&a,&b,&c);
c1=((a>=1) && (a<=10));
c2=((b>=1) && (b<=10));
c3=((c>=1) && (c<=10));
if(!c1)
printf("value of a is out of range"); if(!c2)
printf("value of b is out of range");
if(!c3)
printf("value of c is out of range");
}while(!c1 || !c2 || !c3); if((a+b)>c &&
(b+c)>a && (c+a)>b)
{
if(a==b && b==c)
printf("Triangle is equilateral\n"); else
if(a!=b && b!=c && c!=a)
printf("Triangle is scalene\n"); else
printf("Triangle is isosceles\n");
}
else
printf("Triangle cannot be formed \n");
getch( );
return 0;
}
```

4) Program 4:

Next Date Problem Testing Technique: Equivalence Class Testing 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 test cases and discuss the test results.

```
#include<stdio.h>
int check(int day,int month)
{
if((month==4 || month==6 || month==9 || month==11) && day==31) return 1;
```

```

else
}
return 0;
int isleap(int year)
{
if((year%4==0 && year%100!=0) || year%400==0) return 1;
else
return 0;
}
int main()
{
int day,month,year,tomm_day,tomm_month,tomm_year; char flag;
do
{
flag='y';
printf("\nenter the today's date in the form of dd mm yyyy\n");
scanf("%d%d%d",&day,&month,&year);
tomm_month=month;
tomm_year= year;
if(day<1 || day>31)
{
printf("value of day, not in the range 1...31\n"); flag='n';
}
if(month<1 || month>12)
{
printf("value of month, not in the range 1. 12\n");
flag='n';
}
else if(check(day,month))
{
printf("value of day, not in the range day<=30"); flag='n';
}
if(year<=1812 || year>2015)
{
printf("value of year, not in the range 1812. 2015\n");
flag='n';
}
if(month==2)
{
if(isleap(year) && day>29)
{
printf("invalid date input for leap year");
flag='n';
}
else if(!isleap(year)&& day>28)
{
printf("invalid date input for not a leap year"); flag='n';
}
}
}

```

```
}  
}while(flag=='n');  
switch (month)  
{  
case 1:  
case 3:  
case 5:  
case 7:  
case 8:  
case 10:if(day<31)  
tomm_day=day+1;  
else  
{  
}  
t  
o  
m  
m  
-  
d  
a  
y  
=  
1  
;  
t  
o  
m  
m  
-  
m  
o  
n  
t  
h  
=  
m  
o  
n  
t  
h  
+  
1  
;  
break; case  
4:  
case 6:  
case 9:
```

```

case 11: if(day<30)
tomm_day=day+1;
else
{
tomm_day=1; tomm_month=month+1;
}
break;
case 12: if(day<31)
tomm_day=day+1;
else
{
tomm_day=1;
tomm_month=1;
if(year==2015)
{
printf("the next day is out of boundary value of year\n");
tomm_year=year+1;
}
else
tomm_year=year+1;
}
break;
case 2:
if(day<28)
tomm_day=day+1;
else if(isleap(year)&& day==28)
tomm_day=day+1; else
if(day==28 || day==29)
{
tomm_day=1;
tomm_month=3;
}
break;
}
printf("next day is : %d %d %d",tomm_day,tomm_month,tomm_year);
return 0;

```

5) Program 5

Commission Problem Testing Technique: Boundary Value Analysis 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.

```

#include<stdio.h>
#include<conio.h>
int main()
{

```

```

int c1,c2,c3,temp;
int locks,stocks,barrels,totallocks,totalstocks,totalbarrels;
float lockprice,stockprice,barrelprice,locksales,stocksales,barrelsales,sales,com;
lockprice=45.0;
stockprice=30.0;
barrelprice=25.0;
totallocks=0;
totalstocks=0;
totalbarrels=0; clrscr();
printf("Enter the number of locks and to exit press -1\n");
scanf("%d",&locks);
while(locks != -1)
{
c1=(locks<=0 || locks>70);
printf("\nEnter the number of stocks and barrels\n"); scanf("%d
%d",&stocks,&barrels);
c2=(stocks<=0 || stocks>80);
c3=(barrels<=0 || barrels>90);
if(c1)
printf("\nValue of locks are not in the range of 1. 70\n");
else
{
temp=totallocks+locks; if(temp>70)
printf("New totallocks = %d not in the range of 1. 70\n",temp);
else
totallocks=temp;
}
printf("Total locks = %d",totallocks); if(c2)
printf("\n Value of stocks not in the range of 1. 80\n");
else
{
temp=totalstocks+stocks;
if(temp>80)
printf("\nNew total stocks = %d not in the range of 1. 80",temp);
else
totalstocks=temp;
}
printf("\nTotal stocks = %d",totalstocks); if(c3)
printf("\n Value of barrels not in the range of 1. 90\n");
else
{
temp=totalbarrels+barrels;
if(temp>90)
printf("\nNew total barrels = %d not in the range of 1. 90\n",temp);
else
totalbarrels=temp;
}
printf("\nTotal barrels=%d", totalbarrels);

```

```

printf("\nEnter the number of locks and to exit press -1\n");
scanf("%d",&locks);
}
printf("\n Total locks = %d",totallocks); printf("\n Total stocks = %d",totalstocks); printf("\n Total barrels = %d",totalbarrels);
locksales=totallocks*lockprice;
stocksales=totalstocks*stockprice;
barrelsales=totalbarrels*barrelprice;
sales=locksales+stocksales+barrelsales;
printf("\n Total sales = %f",sales);
if(sales>1800)
{
com=0.10*1000;
com=com+(0.15*800);
com=com+0.20*(sales-1800);
}
else if(sales>1000)
{
com=0.10*1000;
com=com+0.15*(sales-1000);
}
else
com=0.10*sales;
printf("\nCommission = %f",com);
getch();
return 0;
}

```

6) Program 6

Commission Problem Testing Technique: Decision Table based Testing Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.

```

#include<stdio.h>
#include<conio.h>
int main()
{
int c1,c2,c3,temp;
int locks,stocks,barrels,totallocks,totalstocks,totalbarrels;
float lockprice,stockprice,barrelprice,locksales,stocksales,barrelsales,sales,com;
lockprice=45.0;
stockprice=30.0;
barrelprice=25.0;
totallocks=0;
totalstocks=0;
totalbarrels=0; clrscr();
printf("Enter the number of locks and to exit press -1\n");

```

```

scanf("%d",&locks);
while(locks != -1)
{
c1=(locks<=0 || locks>70);
printf("\nEnter the number of stocks and barrels\n"); scanf("%d
%d",&stocks,&barrels);
c2=(stocks<=0 || stocks>80);
c3=(barrels<=0 || barrels>90);
if(c1)
printf("\nValue of locks are not in the range of 1. 70\n");
else
{
temp=totallocks+locks; if(temp>70)
printf("New totallocks = %d not in the range of 1. 70\n",temp);
else
totallocks=temp;
}
printf("Total locks = %d",totallocks); if(c2)
printf("\n Value of stocks not in the range of 1. 80\n");
else
{
temp=totalstocks+stocks;
if(temp>80)
printf("\nNew total stocks = %d not in the range of 1. 80",temp);
else
totalstocks=temp;
}
printf("\nTotal stocks = %d",totalstocks); if(c3)
printf("\n Value of barrels not in the range of 1. 90\n");
else
{
temp=totalbarrels+barrels;
if(temp>90)
printf("\nNew total barrels = %d not in the range of 1. 90\n",temp);
else
totalbarrels=temp;
}
printf("\nTotal barrels=%d", totalbarrels);
printf("\nEnter the number of locks and to exit press -1\n");
scanf("%d",&locks);
}
printf("\n Total locks = %d",totallocks); prin
tf("\n Total stocks = %d",totalstocks); printf("\n
Total barrels = %d",totalbarrels);
locksales=totallocks*lockprice;
stocksales=totalstocks*stockprice;
barreلسales=totalbarrels*barrelprice;
sales=locksales+stocksales+barreلسales;

```

```

printf("\n Total sales = %f",sales);
if(sales>1800)
{
com=0.10*1000;
com=com+(0.15*800);
com=com+0.20*(sales-1800);
}
else if(sales>1000)
{
com=0.10*1000;
com=com+0.15*(sales-1000);
}
else
com=0.10*sales;
printf("\nCommission = %f",com);
getch();
return 0;
}

```

7) Program 7:

Commission Problem Testing Technique: Data Flow Testing Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.

```

1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5 int c1,c2,c3,temp;
6 int locks,stocks,barrels,totallocks,totalstocks,totalbarrels;
7 float lockprice,stockprice,barrelprice,locksales,stocksales,barrelsales,sales,com;
8 lockprice=45.0;
9 stockprice=30.0;
10 barrelprice=25.0;
11 totallocks=0;
12 totalstocks=0;
13 totalbarrels=0;
14 clrscr();
15 printf("Enter the number of locks and to exit press -1\n");
16 scanf("%d",&locks);
17 while(locks != -1)
18 {
19 c1=(locks<=0 || locks>70);
20 printf("\nEnter the number of stocks and barrels\n");
21 scanf("%d %d",&stocks,&barrels);
22 c2=(stocks<=0 || stocks>80);
23 c3=(barrels<=0 || barrels>90);
24 if(c1)
25 printf("\nValue of locks are not in the range of 1. 70\n");

```



```

26 else
27 {
28 temp=totallocks+locks;
29 if(temp>70)
30 printf("New totallocks = %d not in the range of 1. 70\n",temp);
31 else
32 totallocks=temp;
33 }
34 printf("Total locks = %d",totallocks);
35 if(c2)
36 printf("\n Value of stocks not in the range of 1. 80\n");
37 else
38 {
39 temp=totalstocks+stocks;
40 if(temp>80)
41 printf("\nNew total stocks = %d not in the range of 1. 80",temp);
42 else
43 totalstocks=temp;
44 }
45 printf("\nTotal stocks = %d",totalstocks);
46 if(c3)
47 printf("\n Value of barrels not in the range of 1. 90\n");
48 else
49 {
50 temp=totalbarrels+barrels;
51 if(temp>90)
52 printf("\nNew total barrels = %d not in the range of 1. 90\n",temp);
53 else
54 totalbarrels=temp;
55 }
56 printf("\nTotal barrels=%d", totalbarrels);
57 printf("\nEnter the number of locks and to exit press -1\n");
58 scanf("%d",&locks);
59 }
60 printf("\n Total locks = %d",totallocks);
61 printf("\n Total stocks = %d",totalstocks);
62 printf("\n Total barrels = %d",totalbarrels);
63 locksales=totallocks*lockprice;
64 stocksales=totalstocks*stockprice;
65 barrelsales=totalbarrels*barrelprice;
66 sales=locksales+stocksales+barrelsales;
67 printf("\n Total sales = %f",sales);
68 if(sales>1800)
69 {
70 com=0.10*1000;
71 com=com+(0.15*800);
72 com=com+0.20*(sales-1800);
73 }

```

```

74 else if(sales>1000)
75 {
76 com=0.10*1000;
77 com=com+0.15*(sales-1000);
78 }
79 else
80 com=0.10*sales;
81 printf("\nCommission = %f",com);
82 getch();
83 return 0;

```

8) Program 8

Binary Search Testing Technique: Basis Path Testing Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

```

#include<stdio.h>
int binsrc(int x[],int low,int high,int key)
{
    int mid;
    while(low<=high){
        mid=(low+high)/2;
        if(x[mid]==key)
            return mid;
        else if(x[mid]<key)
        {
            low=mid+1;
        }
        else
            high=mid-1;
    }
    return -1;
}
int main()
{
    int a[20],key,i,n,succ;
    printf("Enter the n value up to max of 20"); scanf("%d",&n);
    if(n>0)
    {
        printf("enter the elements in ascending order\n"); for(i=0;i<n;i++)
        scanf("%d",&a[i]);
        printf("enter the key element to be searched\n"); scanf("%d",&key);
        succ=binsrc(a,0,n-1,key); if(succ>=0)
        printf("Element found in position = %d\n",succ+1); else
        printf("Element not found \n");
    }
    else
        printf("Number of element should be greater than zero\n"); return 0;
}

```