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
                                  {
                                  tomm_day=1;
                                  }
```

```
t                                                                   =
o                                                                   m
m                                                                   o
m                                                                   n
_                                                                   t
m                                                                   h
o                                                                   +
n                                                                   1
t                                                                   ;
h
                    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;
```

Equivalence class testing for next date program

Equivalence Classes are as follows:

D1= { Day/DD : 1<=DD<=31 }

M1= { Month/MM : 1<=MM<=12 } Y1= {

Year /YY: 1812<=YY<=2015 }

Weak Normal /Strong Normal

Test cases	Description	Inputs			Output	Comments
		DD	MM	YY		
WN/SN1	Enter valid values for day, month and year from equivalence classes.	12	2	1990	13/2/1990	Valid

Weak Robust

Test cases	Description	Inputs			Output	Comments
		DD	MM	YY		
WR1	Enter valid values for month and year from equivalence classes and invalid value for day.	-1	6	1992	Day out of range	Valid
WR2	Enter valid values for day and year from equivalence classes and invalid value for month.	15	-1	1992	Month out of range	Valid
WR3	Enter valid values for day and month from equivalence classes and invalid value for year.	15	6	1811	Year out of range	Valid
WR4	Enter valid values for month and year from equivalence classes and invalid value for day.	32	6	1992	Day out of range	Valid
WR5	Enter valid values for day and year from equivalence classes and invalid value for month.	15	13	1992	Month out of range	Valid
WR6	Enter valid values for day and month from equivalence classes and invalid value for year.	15	6	2016	Year out of range	Valid

Strong Robust

Test cases	Description	Inputs			Output	Comments
		DD	MM	YY		
SR1	Enter valid values for month and year from equivalence classes and invalid value for day.	-1	6	1992	Day out of range	Valid
SR2	Enter valid values for day and year from equivalence classes and invalid value for month.	15	-1	1992	Month out of range	Valid
SR3	Enter valid values for day and month from equivalence classes and invalid value for year.	15	6	1811	Year out of range	Valid

SR4	Enter valid value for year from equivalence classes and invalid values for day and month.	-1	-1	1992	Day, Month out of range	Valid
SR5	Enter valid value month for from equivalence classes and invalid values for day and year.	-1	6	1811	Day, Year out of range	Valid
SR6	Enter valid value for day from equivalence classes and invalid values for month and year.	15	-1	1811	Month, Year out of range	Valid
SR7	Enter valid values for month and year from equivalence classes and invalid value for day.	32	6	1992	Day out of range	Valid
SR8	Enter valid values for day and year from equivalence classes and invalid value for month.	15	13	1992	Month out of range	Valid
SR9	Enter valid values for day and month from equivalence classes and invalid value for year.	15	6	2016	Year out of range	Valid
SR10	Enter valid value for year from equivalence classes and invalid values for day and month.	32	13	1992	Day, Month out of range	Valid
SR11	Enter valid value month for from equivalence classes and invalid values for day and year.	32	6	2016	Day, Year out of range	Valid
SR12	Enter valid value for day from equivalence classes and invalid values for month and year.	15	13	2016	Month, Year out of range	Valid
SR13	Enter invalid values for day, month and year.	-1	-1	1811	Day, Month, Year out of range	Valid
SR14	Enter invalid values for day, month and year.	32	13	2016	Day, Month, Year out of range	Valid