

Consideremos este código:

```
#include <iostream>
#include <sstream>
#include <stdlib.h>
#include <math.h>

using namespace std;
using std::cout;
using std::endl;
void lectura(int* , int*, int*, int*, int*);
int validar_fecha(int, int, int);
/*****/
int main()
{
    int vd, vm, va, tm = 0, b=0, valido = 0;

    lectura(&vd , &vm , &tm , &va , &b);
    cout<<"\nFecha leida (dd/mm/aaaa): "<<vd<<" / "<<vm<<" / "<<va<<endl;

    valido = validar_fecha(vd , tm , b);

    if(valido==0) { cout<<"FECHA VALIDA"<<endl; }
    else { cout<<"FECHA NO VALIDA"<<endl; }

    cout<<"PROGRAMA TERMINADO"<<endl;
    return 0;
}
/*****/

void lectura(int *d , int *m, int *tm , int *A, int *bis)
{
    int sec1 = 0, sec2 = 0 , b = 0;

    cout<<" - L E C T U R A - "<<endl;

    cout<<"Digite DIA (d)  : "; cin>>*d;
    while( (*d<1) || (*d>31) )
    {
        cout<<"Ningún mes pasa de 31 dias"<<endl;
        cout<<"Digite DIA (d)  : "; cin>>*d;
    }

    cout<<"Digite MES (m)  : "; cin>>*m;
    while( (*m<1) || (*m>12) )
    {
        cout<<"El valor del mes debe estar entre 1 y 12 "<<endl;
        cout<<"Digite MES (m)  : "; cin>>*m;
    }

    while ((*m==2) && (*d>29))
    {
        cout<<"Ningún FEBRERO pasa de 29 dias"<<endl;
        cout<<"Digite DIA (d)  : "; cin>>*d;
    }

    cout<<"Digite ANYO (A) : "; cin>>*A;
    while(*A<1583)
    {
        cout<<"El valor del ANYO debe ser mayor a 1582 "<<endl;
        cout<<"Digite ANYO (A) : "; cin>>*A;
    }

    sec1 = *A%400; //cout<<"sec1 = "<<sec1<<endl;
    sec2 = *A%100; //cout<<"sec2 = "<<sec2<<endl;

    if (sec1==0) *bis = 1;
    else
    {
        b = *A%4;
    }
}
```

```
    if ((sec2!=0) && (b==0)) *bis = 1;
}

switch(*m)
{
case 1:
case 3:
case 5:
case 7:
case 8:
case 10:
case 12: *tm = 1; break;
case 2: *tm = 2; break;
case 4:
case 6:
case 9:
case 11: *tm = 3; break;
}
}
/*****/

int validar_fecha(int dd, int tmm, int bis)
{
    int f_valida = 0;

    if((tmm==2) && (dd>28) && (bis == 0))
        { f_valida = 1; /*cout<<"Dia no valido"<<endl;*/ }

    if( (tmm==3) && (dd>30) )
        { f_valida = 1; /*cout<<"Dia no valido"<<endl;*/ }

    return f_valida;
}
/*****/
```

Efectúe cobertura de condición múltiple para el subprograma:

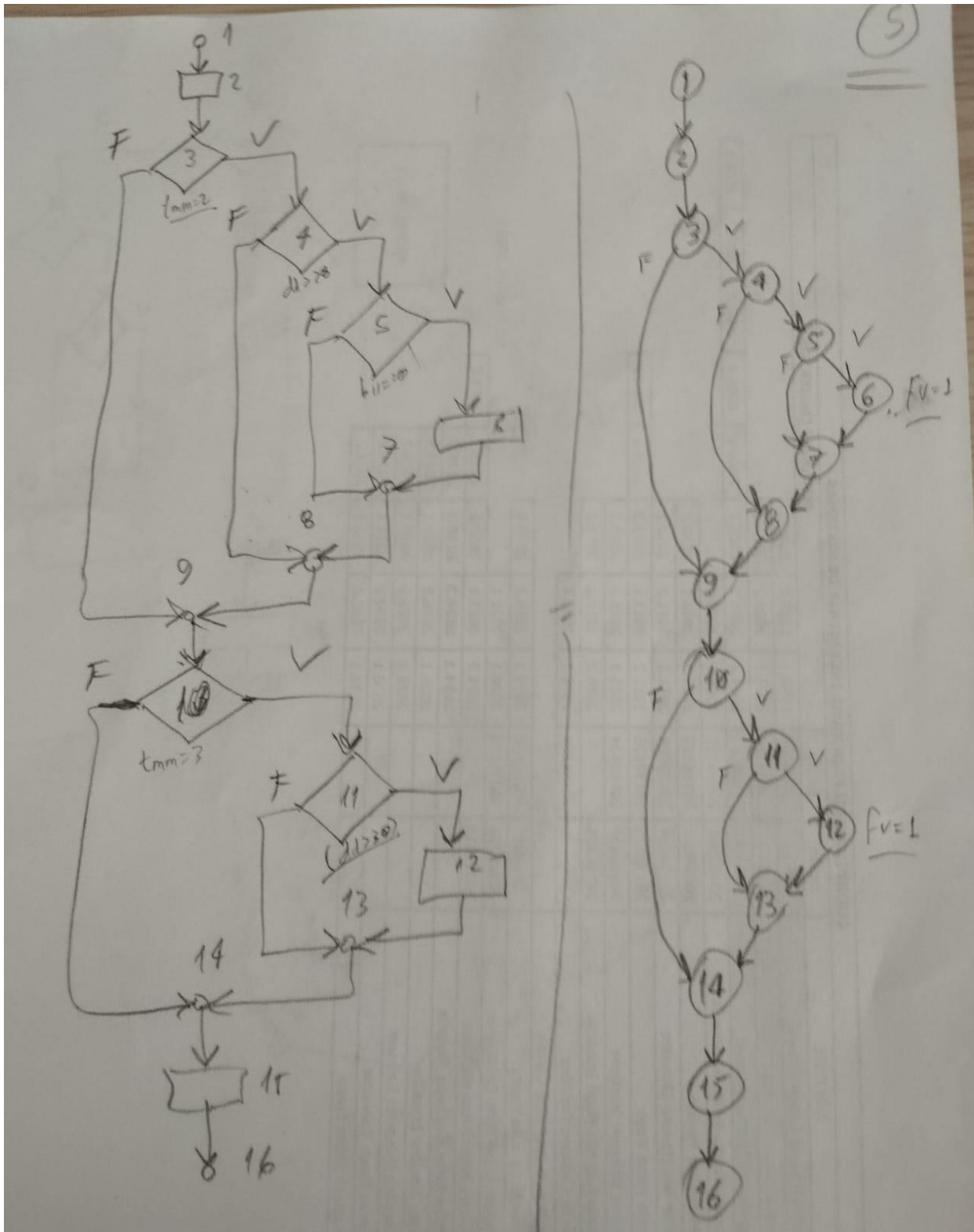
validar_fecha

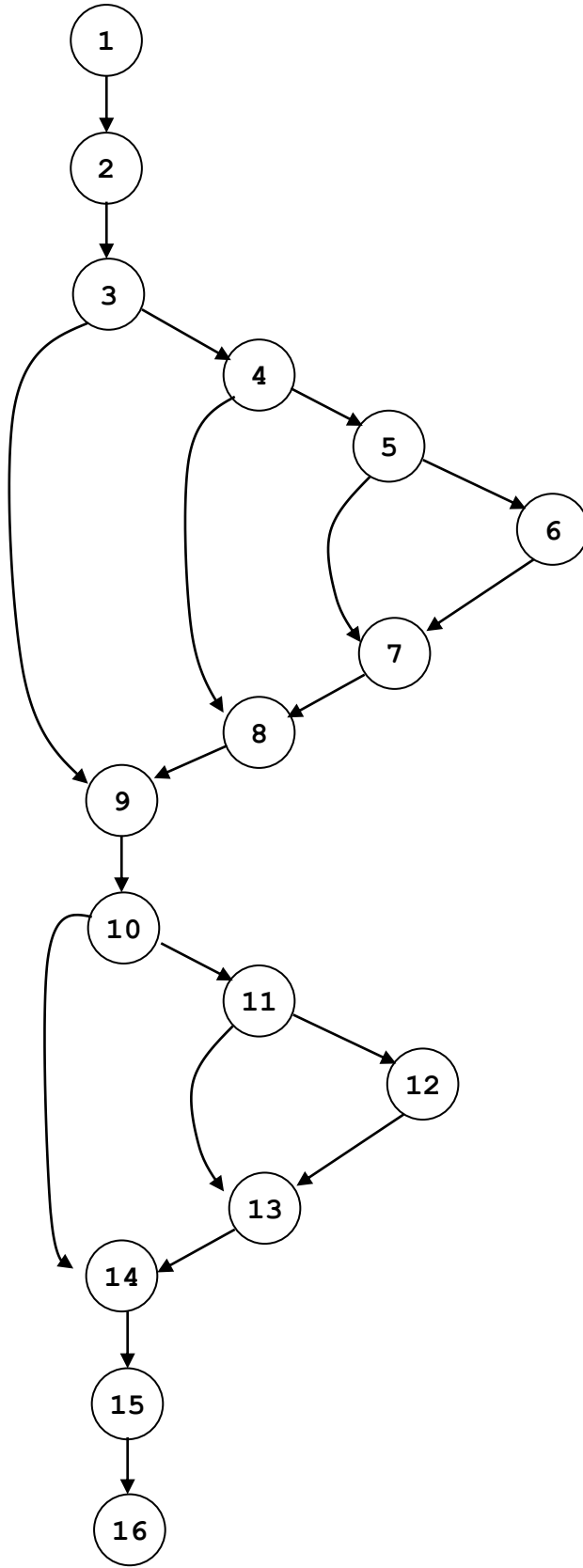
Debe entregar:

- Diagrama de flujo y Grafo con los nodos numerados.
- El análisis de combinaciones teóricas
- Cuáles de las anteriores representan caminos que serán casos de prueba
- Indique valores para los casos de prueba

SOLUCIÓN:

a. Diagrama de flujo y Grafo con los nodos numerados.





b. El análisis de combinaciones teóricas

Comb	E1 (tmm==2)	E2 (dd>28)	E3 (bis == 0)	E4 (tmm==3)	E5 (dd>30)	E1 && E2 && E3	E4 && E5	(E1 && E2 && E3) OR (E4 && E5)	CAMINOS
1	V	V	V	V	V	V	V	V	NO
2	V	V	V	V	F	V	F	V	NO
3	V	V	V	F	V	V	F	V	1
4	V	V	V	F	F	V	F	V	1
5	V	V	F	V	V	F	V	V	NO
6	V	V	F	V	F	F	F	F	NO
7	V	V	F	F	V	F	F	F	2
8	V	V	F	F	F	F	F	F	2
9	V	F	V	V	V	F	V	V	NO
10	V	F	V	V	F	F	F	F	NO
11	V	F	V	F	V	F	F	F	3
12	V	F	V	F	F	F	F	F	3
13	V	F	F	V	V	F	V	V	NO
14	V	F	F	V	F	F	F	F	NO
15	V	F	F	F	V	F	F	F	3
16	V	F	F	F	F	F	F	F	3
17	F	V	V	V	V	F	V	V	4
18	F	V	V	V	F	F	F	F	5
19	F	V	V	F	V	F	F	F	6
20	F	V	V	F	F	F	F	F	6
21	F	V	F	V	V	F	V	V	4
22	F	V	F	V	F	F	F	F	5
23	F	V	F	F	V	F	F	F	6
24	F	V	F	F	F	F	F	F	6
25	F	F	V	V	V	F	V	V	4
26	F	F	V	V	F	F	F	F	5
27	F	F	V	F	V	F	F	F	6
28	F	F	V	F	F	F	F	F	6
29	F	F	F	V	V	F	V	V	4
30	F	F	F	V	F	F	F	F	5
31	F	F	F	F	V	F	F	F	6
32	F	F	F	F	F	F	F	F	6

c. Cuáles de las anteriores representan caminos que serán casos de prueba

NO POSIBLES: Combinaciones: 1, 2, 5, 6, 9, 10, 13, 14

CAMINOS (CASOS DE PRUEBA):

Camino 1: Combinaciones 3 , 4

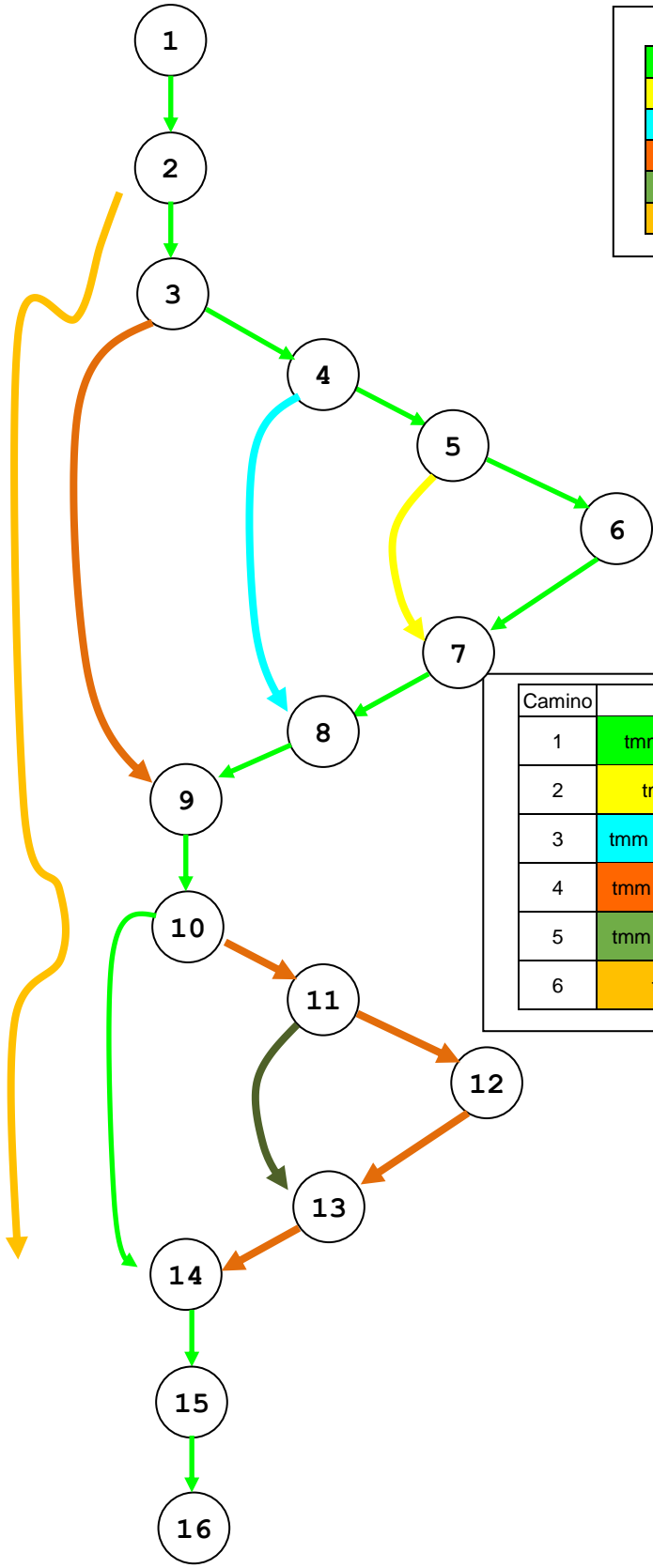
Camino 2: Combinaciones 7 , 8

Camino 3: Combinaciones 11 , 12 , 15 , 16

Camino 4: Combinaciones 17 , 21 , 25 , 29

Camino 5: Combinaciones 18 , 22 , 26 , 30

Camino 6: Combinaciones 19 , 20 , 23, 24, 27 , 28, 31, 32



- C1: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 14 - 15 - 16
- C2: 1 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 10 - 14 - 15 - 16
- C3: 1 - 2 - 3 - 4 - 8 - 9 - 10 - 14 - 15 - 16
- C4: 1 - 2 - 3 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16
- C5: 1 - 2 - 3 - 9 - 10 - 11 - 13 - 14 - 15 - 16
- C6: 1 - 2 - 3 - 9 - 10 - 14 - 15 - 16

Camino	Situación	Salida Esperada
1	tmm = 2 , dd = 29 , año NO bisiesto	f_valida = 1 (NO VÁLIDA)
2	tmm = 2 , dd = 29 , año bisiesto	f_valida = 0 (FECHA VÁLIDA)
3	tmm = 2 , dd = 28 , año INDIFERENTE	f_valida = 0 (FECHA VÁLIDA)
4	tmm = 3 , dd = 31 , año INDIFERENTE	f_valida = 1 (NO VÁLIDA)
5	tmm = 3 , dd = 30 , año INDIFERENTE	f_valida = 0 (FECHA VÁLIDA)
6	tmm = 1 , dd INDIF , año INDIF	f_valida = 0 (FECHA VÁLIDA)

d. Indique valores para los casos de prueba

Camino	Situación	Salida Esperada
1	tmm = 2 , dd = 29 , año NO bisiesto (tmm==2) AND (dd>28) AND (bis==0). (IMPLICA (tmm= 3): FALSO)) (VERD) AND (VERD) AND (VERD) ENTRADAS: validar_fecha(29 , 2 , 0)	f_valida = 1 (FECHA NO VÁLIDA)
2	tmm = 2 , dd = 29 , año bisiesto (tmm==2) AND (dd>28) AND (bis==0). (IMPLICA (tmm= 3): FALSO)) (VERD) AND (VERD) AND (FALSO) ENTRADAS: validar_fecha(29 , 2 , 1)	f_valida = 0 (FECHA VÁLIDA)
3	tmm = 2 , dd = 28 , año INDIFERENTE (tmm==2) AND (dd>28) AND (bis INDIF). (IMPLICA (tmm= 3): FALSO)) (VERD) AND (FALSO) ENTRADAS: validar_fecha(28 , 2 , 1)	f_valida = 0 (FECHA VÁLIDA)
4	tmm = 3 , dd = 31 , año INDIFERENTE (tmm==3) AND (dd>30). (IMPLICA (tmm= 2): FALSO)) (VERD) AND (VERD) ENTRADAS: validar_fecha(31 , 3 , 0)	f_valida = 1 (FECHA NO VÁLIDA)
5	tmm = 3 , dd = 30 , año INDIFERENTE (tmm==3) AND (dd>30). (IMPLICA (tmm= 2): FALSO)) (VERD) AND (FALSO) ENTRADAS: validar_fecha(30 , 3 , 1)	f_valida = 0 (FECHA VÁLIDA)
6	tmm =1 , dd INDIF, año INDIF (tmm==3) AND (dd>30). (IMPLICA (tmm= 2): FALSO)) (FALSO) AND (dd INDIF). ENTRADAS: validar_fecha(31 , 1 , 0)	f_valida = 0 (FECHA VÁLIDA)

----- FIN DEL DOCUMENTO