

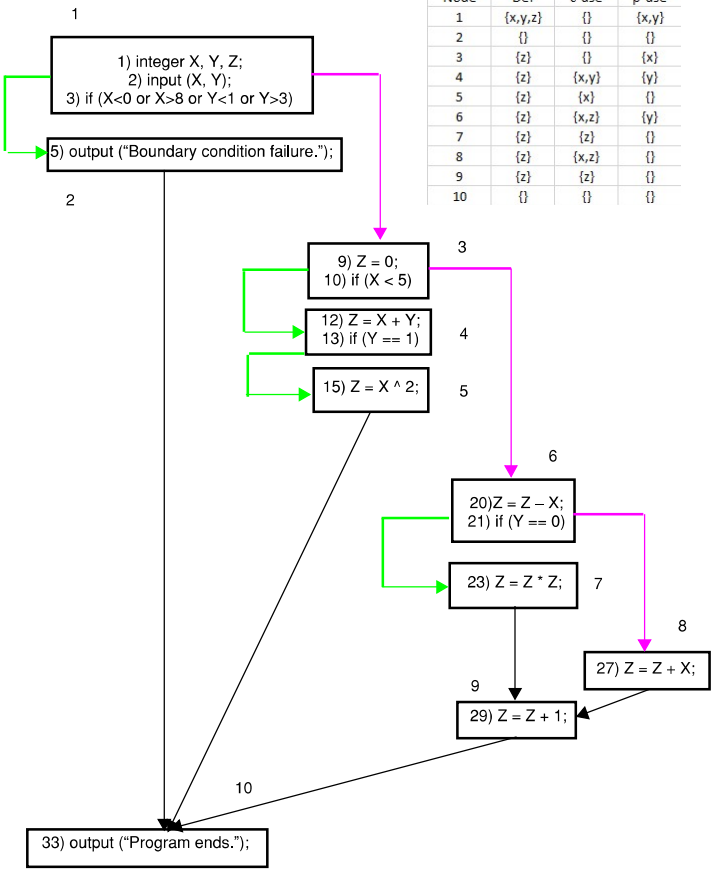
Original Program P

```
1) integer X, Y, Z;  
2) input (X, Y);  
3) if (X<0 or X>8 or Y<1 or Y>3)  
4) {  
5) output ("Boundary condition failure.");  
6) } // end if invalid inputs  
7) else  
8) {  
9)   Z = 0;  
10)  if (X < 5)  
11)  {  
12)    Z = X + Y;  
13)    if (Y == 1)  
14)    {  
15)      Z = X ^ 2;  
16)    } // end if (Y==1)  
17)  } // end if (X<5)  
18)  else  
19)  {  
20)    Z = Z - X;  
21)    if (Y == 0)  
22)    {  
23)      Z = Z * Z;  
24)    } // end if (Y==2)  
25)    else  
26)    {  
27)      Z = Z + X;  
28)    } // end else !(Y==2)  
29)    Z = Z + 1;  
30)  } // end else !(X<5)  
31)  output (X,Y,Z);  
32) } // end else legal inputs  
33) output ("Program ends.");  
34) end;
```

Without Syntactical Markers

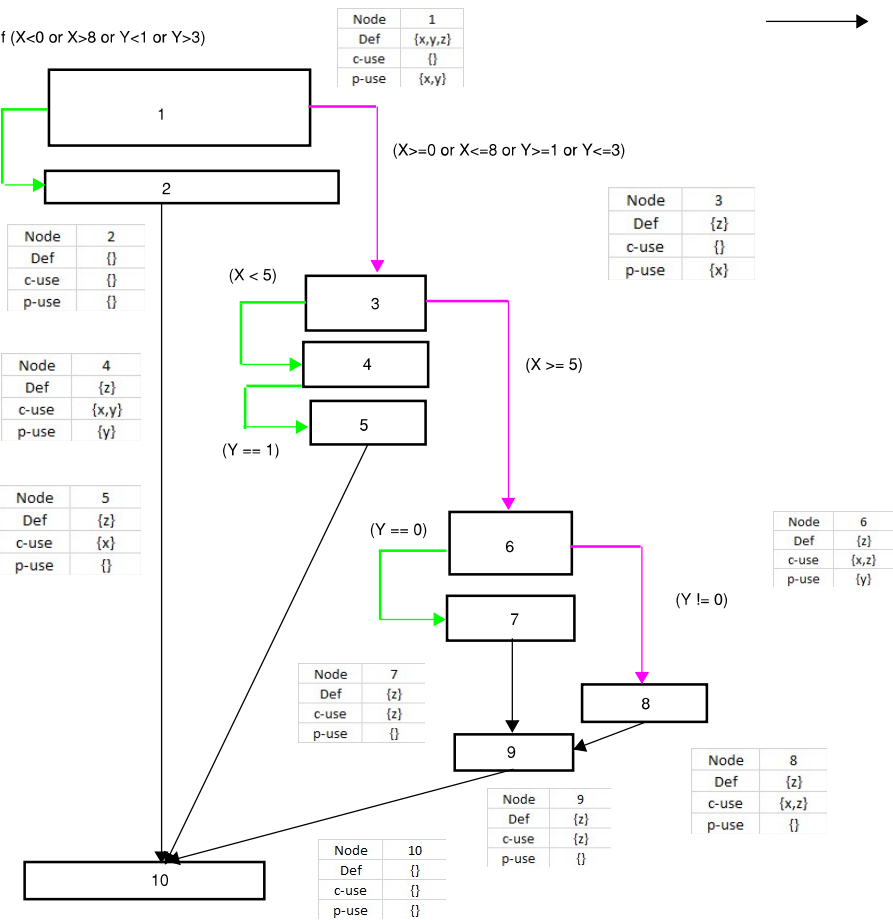
```
1) integer X, Y, Z;  
2) input (X, Y);  
3) if (X<0 or X>8 or Y<1 or Y>3)  
4) {  
5) output ("Boundary condition failure.");  
6) } // end if invalid inputs  
7) else  
8) {  
9)   Z = 0;  
10)  if (X < 5)  
11)  {  
12)    Z = X + Y;  
13)    if (Y == 1)  
14)    {  
15)      Z = X ^ 2;  
16)    } // end if (Y==1)  
17)  } // end if (X<5)  
18)  else  
19)  {  
20)    Z = Z - X;  
21)    if (Y == 0)  
22)    {  
23)      Z = Z * Z;  
24)    } // end if (Y==2)  
25)    else  
26)    {  
27)      Z = Z + X;  
28)    } // end else !(Y==2)  
29)    Z = Z + 1;  
30)  } // end else !(X<5)  
31)  output (X,Y,Z);  
32) } // end else legal inputs  
33) output ("Program ends.");  
34) end;
```

Control Flow Graph



Node	Def	c-use	p-use
1	{x,y,z}	{}	{x,y}
2	{}	{}	{}
3	{z}	{}	{x}
4	{z}	{x,y}	{y}
5	{z}	{x}	{}
6	{z}	{x,z}	{y}
7	{z}	{z}	{}
8	{z}	{x,z}	{}
9	{z}	{z}	{}
10	{}	{}	{}

Data Flow Graph



Node	1
Def	{x,y,z}
c-use	{}
p-use	{x,y}

Node	3
Def	{z}
c-use	{}
p-use	{x}

Node	6
Def	{z}
c-use	{x,z}
p-use	{y}

Node	7
Def	{z}
c-use	{z}
p-use	{}

Node	8
Def	{z}
c-use	{x,z}
p-use	{}

Node	10
Def	{}
c-use	{}
p-use	{}