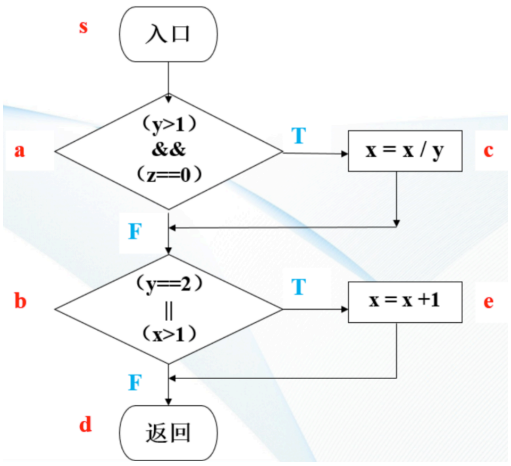
白盒测试

1. 逻辑覆盖案例



1）语句覆盖

测试用例	输入	预期输出	被测路径
CASE1	x=4 y=2 z=0	x=3	sacbed

2）判定覆盖

测试用例	输入	预期输出	覆盖分支	被测路径
CASE2	x=1 y=3 z=0	x=1/3	ac，bd	sacbd
CASE3	x=3 y=2 z=1	X=4	ab，be	sabed

3）条件覆盖

测试用例	输入	预期输出	覆盖条件	覆盖分支	被测路径
CASE4	x=0 y=2 z=0	x=1	T1， T2 T3，-T4	ac， be	sacbed
CASE5	x=2 y=1 z=1	x=3	-T1，-T2 -T3， T4	ab， be	sabed

4）判定/条件覆盖

测试用例	输入	预期输出	覆盖条件	覆盖分支	被测路径
CASE6	x=4，y=2，z=0	x=3	T4，T1 T3，T2	ac，be	sacbed
CASE7	x=1，y=1，z=1	x=1	-T4，-T1 -T3，-T2	ab，bd	sabd

5）条件组合覆盖

编号	判定1各条件组合	编号	判定2各条件组合
1	y>1，z==0	5	y=2，x>1
2	y>1，z!=0	6	y=2，x≤1
3	y<1，z==0	7	y!=2，x>1
4	y<1，z!=0	8	y!=2，x≤1

测试用例	输入	预期输出	覆盖条件组合	被测路径
CASE8	x=4，y=2，z=0	x=3	1，5	sacbed
CASE9	x=1，y=2，z=1	x=2	2，6	sabed
CASE10	x=2，y=1，z=0	x=3	3，7	sabed
CASE11	x=1，y=1，z=1	x=1	4，8	sabd

6）路径覆盖

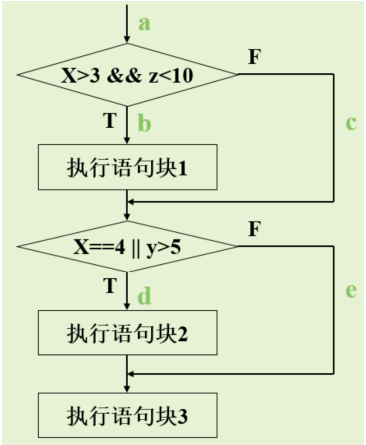
测试用例	输入	预期输出	被测路径
CASE8	x=4，y=2，z=0	x=3	sacbed
CASE9	x=1，y=2，z=1	x=2	sabed
CASE10	x=1，y=3，z=0	x=1/3	sacbd
CASE11	x=1，y=1，z=1	x=1	sabd

2. DoWork

程序如下

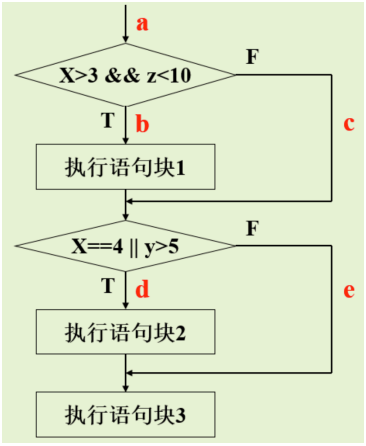
```
void DoWork(int x, int y, int z) {
    int k = 0, j = 0;
    if( (x > 3) && (z < 10) ) {
        k = x * y - 1; // 语句块1
        j = sqrt(k);
    }
    if( (x == 4) || (y > 5) ) {
        j = x * y + 10; // 语句块2
    }
    j = j % 3; // 语句块3
}
```

流程图



1）语句覆盖

输入 `x = 4, y = 5, z = 5` ，程序执行路径 `abd`



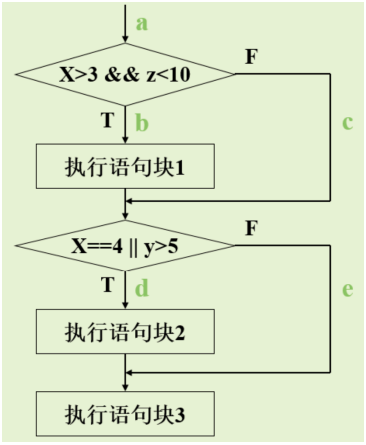
2) 判定覆盖

输入

- `x = 4, y = 5, z = 5`
- `x = 2, y = 5, z = 5`

执行路径

- `abd`
- `ace`

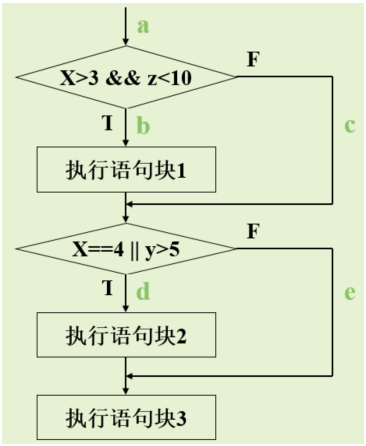


3）条件覆盖

- 对于第一个判定 `((x > 3) && (z < 10))`
  - 条件 `x > 3` 取真值 `T1`，假值 `-T1`
  - 条件 `z < 10` 取真值 `T2`，假值 `-T2`
- 对于第二个判定 `( (x == 4) || (y > 5) )`
  - 条件 `x == 4` 取真值记为 `T3`，假值记为 `-T3`
  - 条件 `y > 5` ，取真值记为 `T4`，假值记为 `-T4`

测试用例	执行路径	覆盖条件	覆盖分支
x=4、y=6、z=5	abd	T1、T2、T3、T4	bd
x=2、y=5、 z=15	ace	-T1、-T2、-T3、-T4	ce

4）判定/条件覆盖



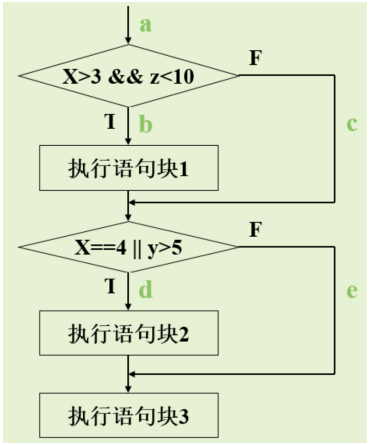
测试用例	执行路径	覆盖条件	覆盖分支
x=4 y=6 z=5	abd	T1、T2、 T3、T4	bd
x=2 y=5 z=15	ace	-T1、-T2、 -T3、-T4	ce

5）条件组合

1、$x>3, z<10$，记做T1, T2，第一个判定的取真分支
2、$x>3, z\geq 10$，记做T1, -T2,第一个判定的取假分支
3、$X\leq 3, z<10$，记做-T1, T2,第一个判定的取假分支
4、$x\leq 3, z\geq 10$，记做-T1, -T2,第一个判定的取假分支
5、$x\leq 4, y>5$，记做T3, T4,第二个判定的取真分支
6、$x==4, y\leq 5$，记做T3, -T4，第二个判定的取真分支
7、$x!=4, y>5$，记做-T3, T4,第二个判定的取真分支
8、$x!=4, y\leq 5$，记做-T3, -T4，第二个判定的取假分支

测试用例	执行路径	覆盖条件	覆盖组合号
x=4、y=6、z=5	abd	T1、T2、 T3、T4	1和5
x=4、y=5、z=15	acd	T1、-T2、 T3、-T4	2和6
x=2、y=6、z=5	acd	-T1、T2、 -T3、T4	3和7
x=2、y=5、z=15	ace	-T1、-T2、 -T3、-T4	4和8

6）路径覆盖



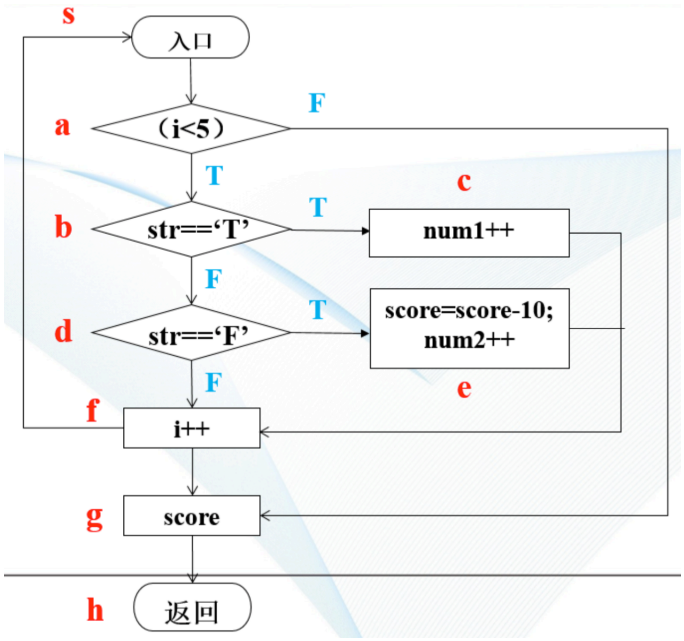
测试用例	执行路径
x=4、y=6、z=5	abd
x=4、y=5、z=15	acd
x=2、y=5、z=15	ace
x=5、y=5、z=5	abe

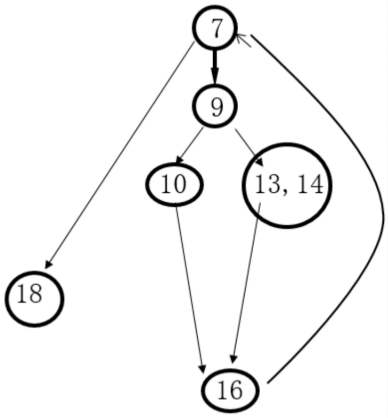
3. 基本路径

1）流图

```
main() {
    int num1 = 0, num2 = 0, score = 100;
    int i;
    char str;
    scanf("%d,%c\n", &i, &str);
    while(i < 5) {
        if(str == 'T')
            num ++;
        else if(str == 'F'){
            score = score - 10;
            num2 ++;
        }
        i ++;
    }
    printf("num1 = %d, num2 = %d, score = %d\n", num1, num2, score);
}
```

程序流图



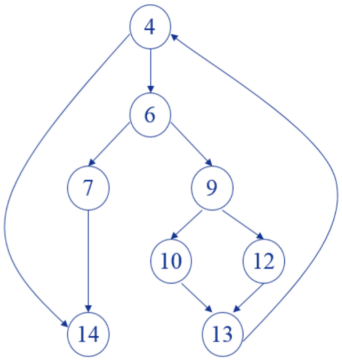


2) sort

```
void Sort(int iRecordNum, int iType) {
    int x = 0, y = 0;
    while(iRecodNum <= 10) {
        if(iType == 0) {
            x = y + 2;
            break;
        }
        else {
            if( iType == 1) {
                x = y + 10;
                iRecordNum ++;
            }
            else {
                x = y + 20;
                iRecordNum ++;
            }
        }
    }
```

```
}
```

控制流图



计算环形复杂度

- $10edges - 8nodes + 2 = 4$
- 路径1：$4 \Longrightarrow 14$
- 路径2：$4 \Longrightarrow 6 \Longrightarrow 7 \Longrightarrow 14$
- 路径3：$4 \Longrightarrow 6 \Longrightarrow 9 \Longrightarrow 10 \Longrightarrow 13 \Longrightarrow 4 \cdots$
- 路径4：$4 \Longrightarrow 6 \Longrightarrow 9 \Longrightarrow 12 \Longrightarrow 13 \Longrightarrow 4 \cdots$

$\cdots$ 表示后面的路径是可以选择的，存在循环

设计测试用例

	输入数据	预期输出
测试用例1	irecordnum = 11	x = ? y = ?
测试用例2	irecordnum = 10 itype = 0	x = ? y = ?
测试用例3	irecordnum = 10 itype = 1	x = ? y = ?
测试用例4	irecordnum = 10 itype = 2	x = ? y = ?