



**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

*Nhóm chuyên môn Nhập môn Công nghệ phần mềm*

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Hướng dẫn bài tập:

**Phương pháp kiểm thử hộp trắng**



1. Bài toán kiểm thử với đồ thị luồng điều khiển
2. Bài tập áp dụng

*Sau bài học này, người học có thể:*

1. Nắm được **phương pháp** giải quyết bài toán kiểm thử với đồ thị luồng điều khiển
2. Biết **vận dụng kiến thức** về kỹ thuật kiểm thử cho bài toán kiểm thử với các tiêu chí bao phủ đường dẫn, bao phủ nhánh và bao phủ câu lệnh

## 1. Bài toán kiểm thử với đồ thị luồng điều khiển

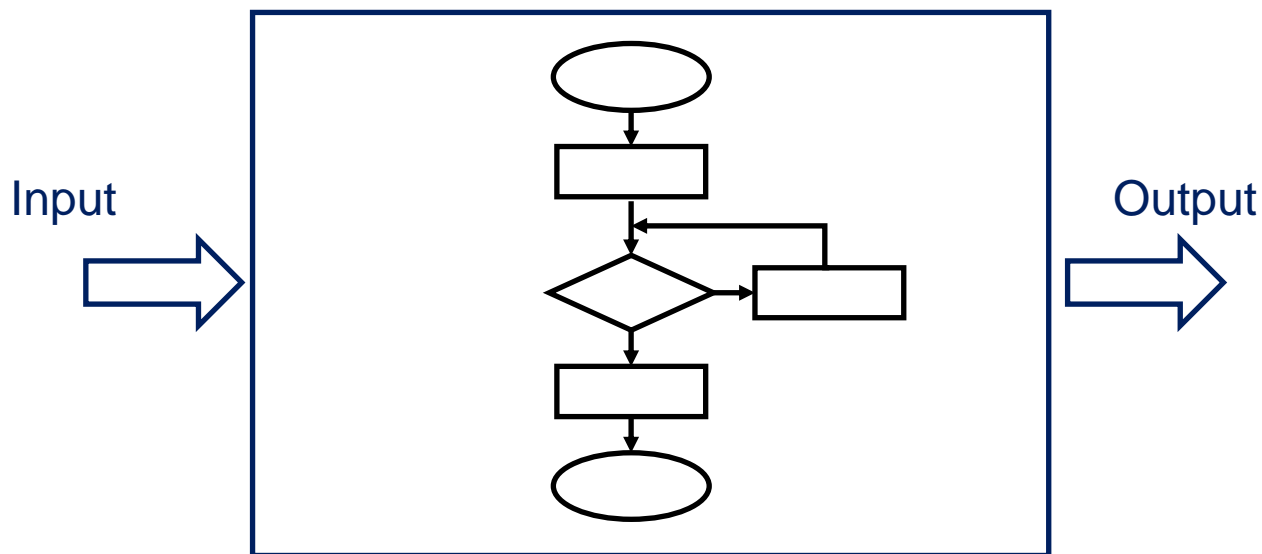
## 2. Bài tập áp dụng

# 1. BÀI TOÁN KIỂM THỬ HỘP TRẮNG

## Bài toán kiểm thử luồng điều khiển

### ■ Mô tả

- Cho một đoạn chương trình (mã nguồn)
- Thiết kế các ca kiểm thử đảm bảo bao phủ đường dẫn (/nhánh/câu lệnh)



Hình 1.1: Kiểm thử hộp trắng

# 1. BÀI TOÁN KIỂM THỬ HỘP TRẮNG

---



## Bài toán kiểm thử luồng điều khiển

### ▪ Cách giải quyết bài toán

- Bước 1: Vẽ đồ thị CFG tương ứng với đoạn chương trình đã cho.
- Bước 2: Xác định các đường đi (/nhánh rẽ/câu lệnh) tương ứng theo yêu cầu của đề bài.
- Bước 3: Xác định các giá trị đầu vào tương ứng theo các đường đi (/nhánh rẽ/câu lệnh) ở bước 2.
- Bước 4: Xác định các test case tương ứng với các giá trị tìm được ở bước 3.

## 2. BÀI TẬP ÁP DỤNG

---



### Ví dụ 1: giảm giá sản phẩm

Cho chương trình để xét giảm giá sản phẩm với mô tả như sau:

- Đầu vào:
  - có là thành viên hay không
  - có mã khuyến mãi hay không
- Đầu ra:
  - giảm 10% khi là thành viên hoặc giảm 7% nếu chưa phải là thành viên
  - giảm thêm 5% nếu có mã khuyến mãi hoặc chỉ giảm 3% nếu không có mã

## 2. BÀI TẬP ÁP DỤNG

---



### Đoạn mã chương trình

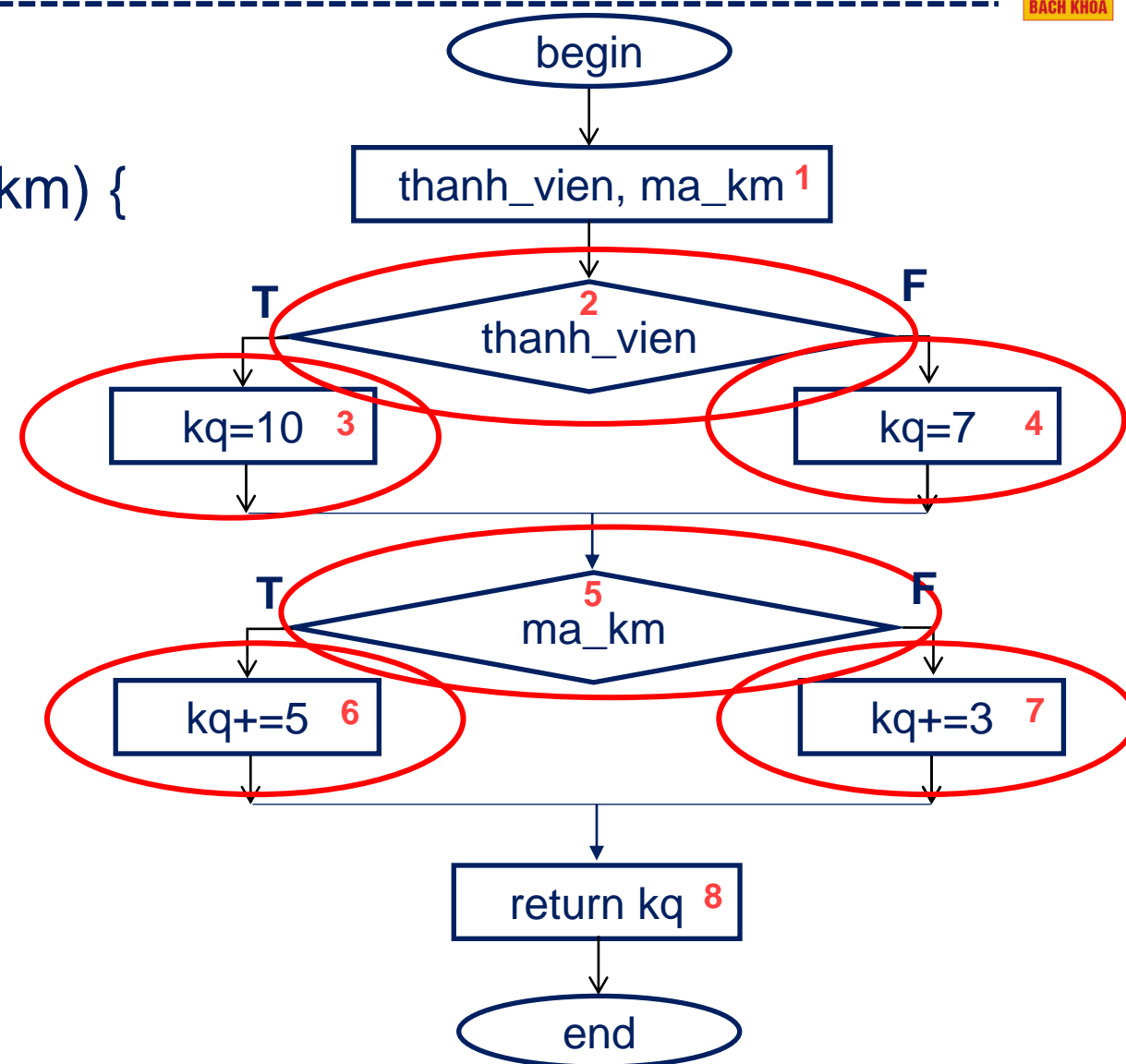
```
int giamgia(bool thanh_vien, bool ma_km) {  
    int kq = 0;  
    if (thanh_vien) kq = 10; else kq = 7;  
    if (ma_km) kq += 5; else kq += 3;  
    return kq;  
}
```



## 2. BÀI TẬP ÁP DỤNG

### Đoạn mã chương trình

```
int giamgia(bool thanh_vien, bool ma_km) {  
    int kq;  
    if (thanh_vien) kq = 10; else kq = 7;  
    if (ma_km) kq += 5; else kq += 3;  
    return kq;  
}
```

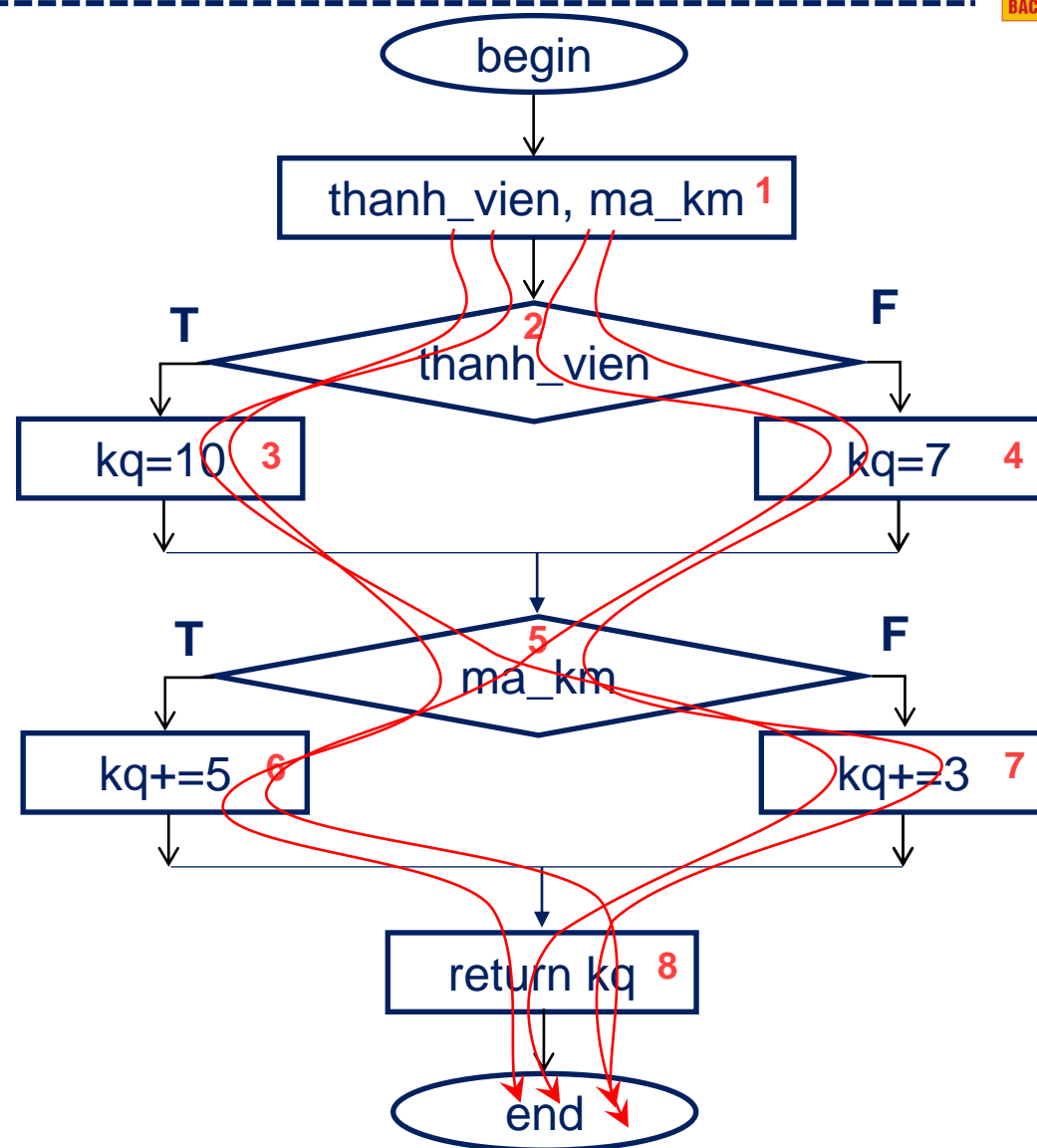


Hình 2.1: Sơ đồ luồng điều khiển – ví dụ 1

## 2. BÀI TẬP ÁP DỤNG

### Phủ đường dẫn

- (1-2-3-5-6-8): **TC <T, T, 15>**
- (1-2-3-5-7-8): **TC <T, F, 13>**
- (1-2-4-5-6-8): **TC <F, T, 12>**
- (1-2-4-5-7-8): **TC <F, F, 10>**



Hình 2.1: Sơ đồ luồng điều khiển – ví dụ 1

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ nhánh (1): **T-T**

```
int giamgia(bool thanh_vien, bool ma_km) {
```

```
    int kq;
```

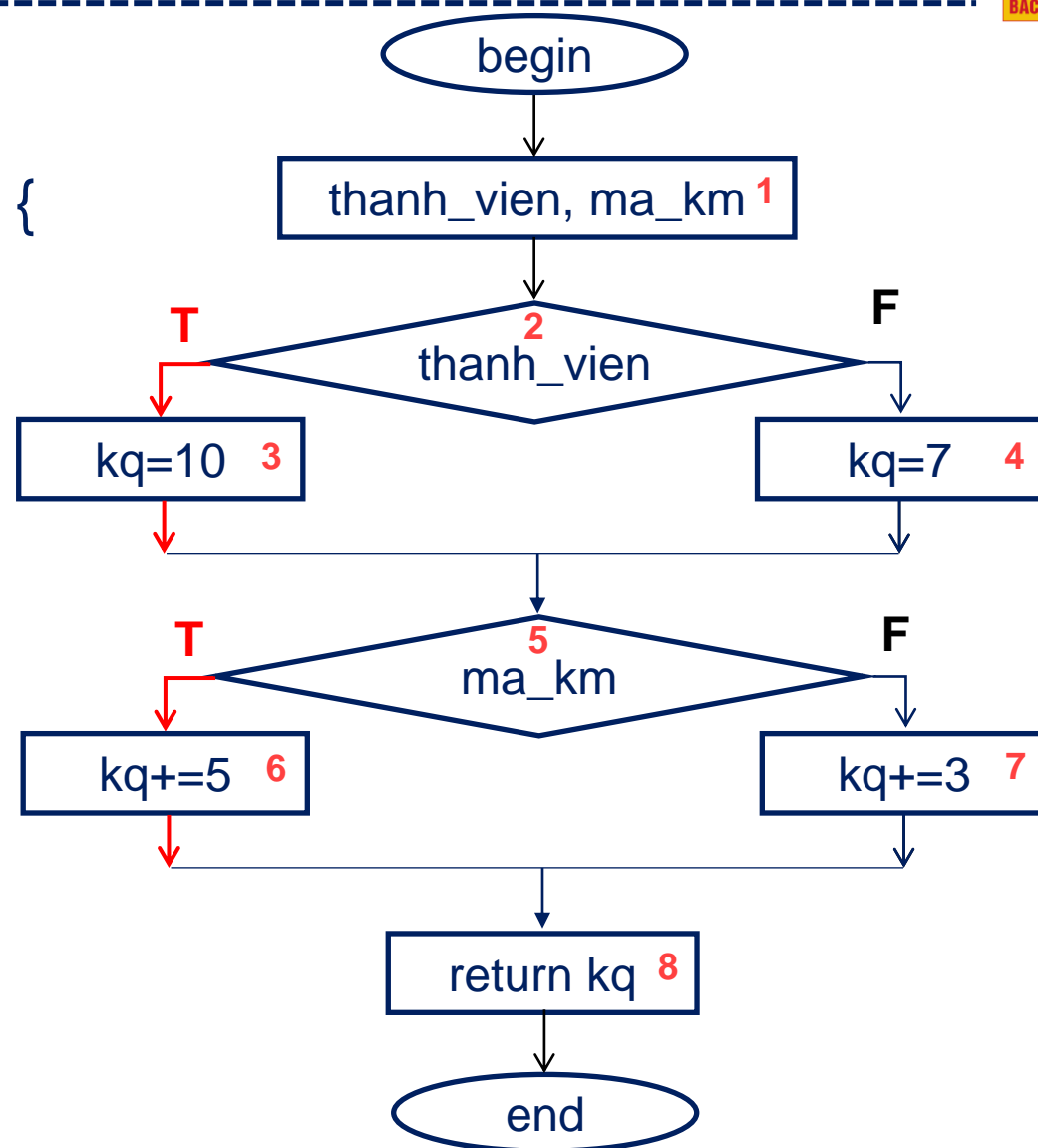
```
    if (thanh_vien) kq = 10; else kq = 7;
```

```
    if (ma_km) kq += 5; else kq += 3;
```

```
    return kq;
```

```
}
```

▪ **TC <T, T, 15>**



Hình 2.2: Bao phủ nhánh T-T

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ nhánh (2): **F-F**

```
int giamgia(bool thanh_vien, bool ma_km) {
```

```
    int kq;
```

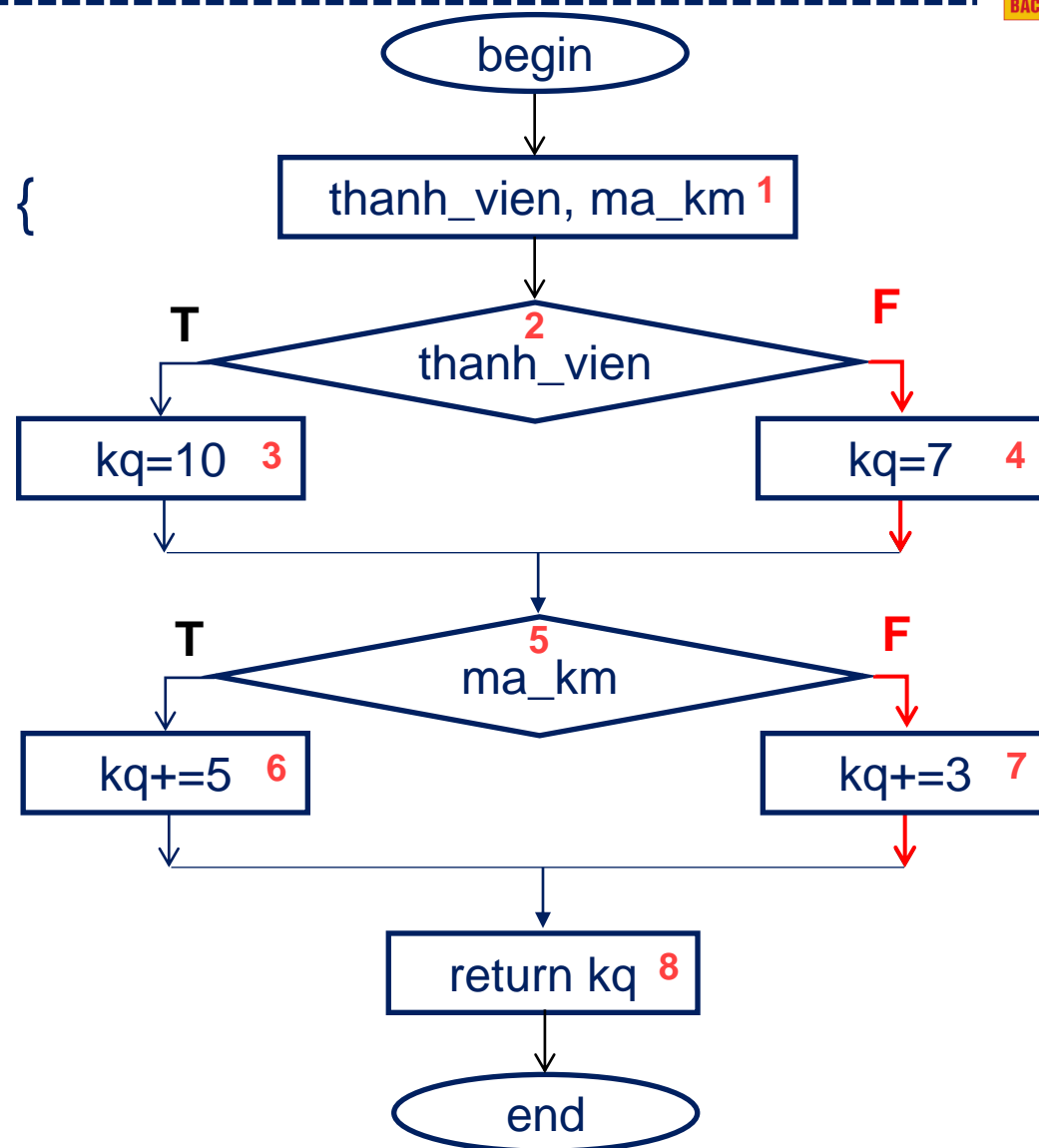
```
    if (thanh_vien) kq = 10; else kq = 7;
```

```
    if (ma_km) kq += 5; else kq += 3;
```

```
    return kq;
```

```
}
```

▪ **TC <F, F, 10>**



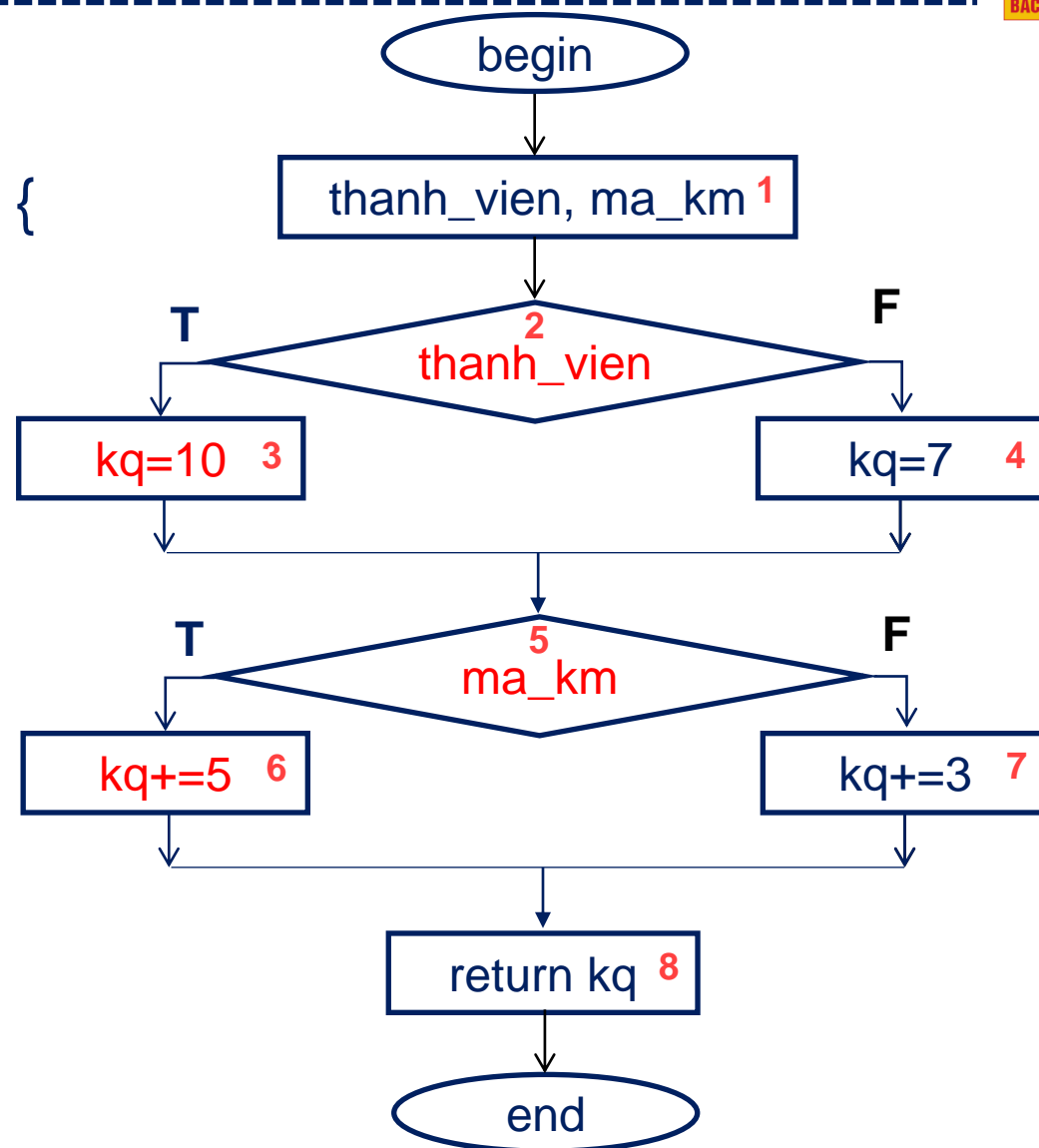
Hình 2.3: Bao phủ nhánh F-F

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ lệnh (1):

```
int giamgia(bool thanh_vien, bool ma_km) {  
    int kq;  
    if (thanh_vien) kq = 10; else kq = 7;  
    if (ma_km) kq += 5; else kq += 3;  
    return kq;  
}
```

▪ TC <T, T, 15>



Hình 2.4: Bao phủ lệnh `kq=10` và `kq+=5`

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ lệnh (2):

```
int giamgia(bool thanh_vien, bool ma_km) {
```

```
    int kq;
```

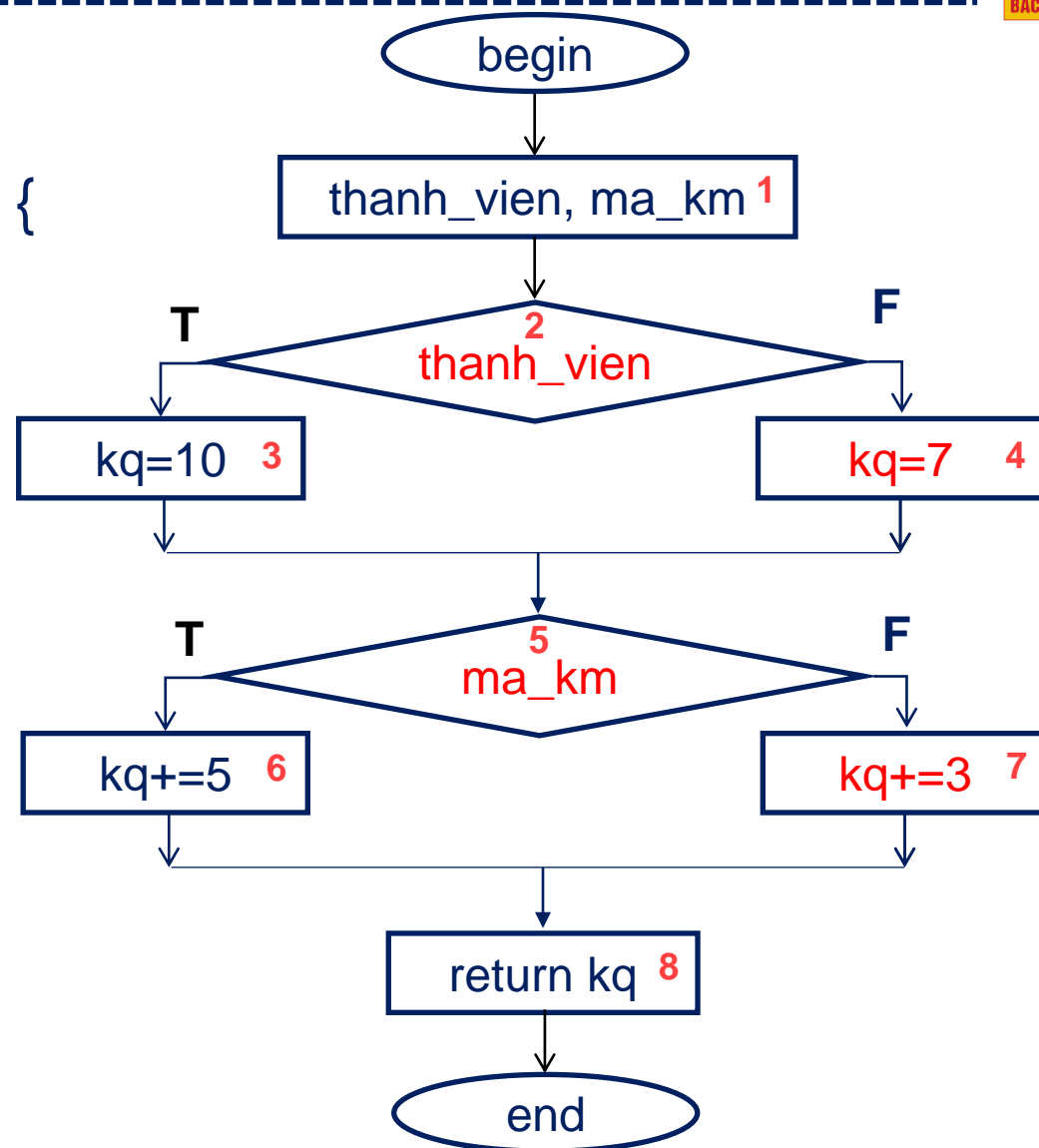
```
    if (thanh_vien) kq = 10; else kq = 7;
```

```
    if (ma_km) kq += 5; else kq += 3;
```

```
    return kq;
```

```
}
```

▪ **TC <F, F, 10>**



Hình 2.5: Bao phủ lệnh `kq=7` và `kq+=3`

## 2. BÀI TẬP ÁP DỤNG

---



### Nhận xét

- Trong bài tập này ta có
  - Bao phủ đường dẫn: 4 test case
  - Bao phủ nhánh: 2 test case
  - Bao phủ lệnh: 2 test case
- Ta thấy tiêu chí bao phủ đường dẫn cần đến 4 test case và là tiêu chí mạnh nhất.

## 2. BÀI TẬP ÁP DỤNG

---



### Ví dụ 2: xét kết quả điểm học phần

Cho chương trình xét điểm học phần của sinh viên với mô tả như sau:

- Đầu vào:
  - điểm quá trình và điểm cuối kì (qt và ck) - giả thiết thuộc khoảng  $[0,10]$
- Đầu ra:
  - ‘đạt’ nếu  $(qt * 0.3 + ck * 0.7) \geq 4$  và không có điểm liệt ( $< 3$ )
  - ‘không đạt’ nếu ngược lại



## 2. BÀI TẬP ÁP DỤNG

---

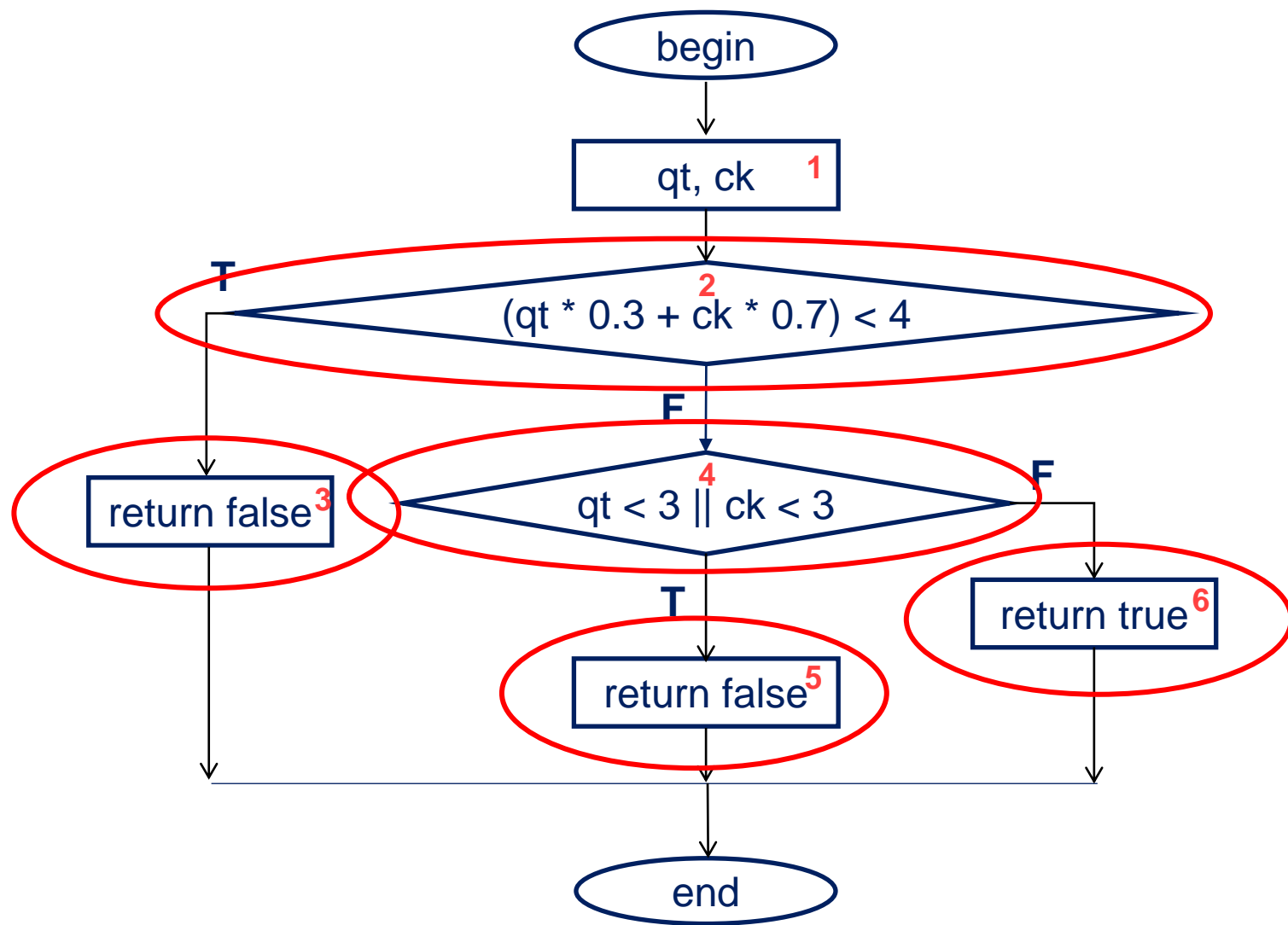
### Đoạn mã chương trình

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + ck * 0.7) < 4)  
        return false;  
    else  
        if (qt < 3 || ck < 3)  
            return false;  
        else  
            return true;  
}
```

## 2. BÀI TẬP ÁP DỤNG

### Đoạn mã chương trình

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + ck * 0.7) < 4)  
        return false;  
    else  
        if (qt < 3 || ck < 3)  
            return false;  
        else  
            return true;  
}
```

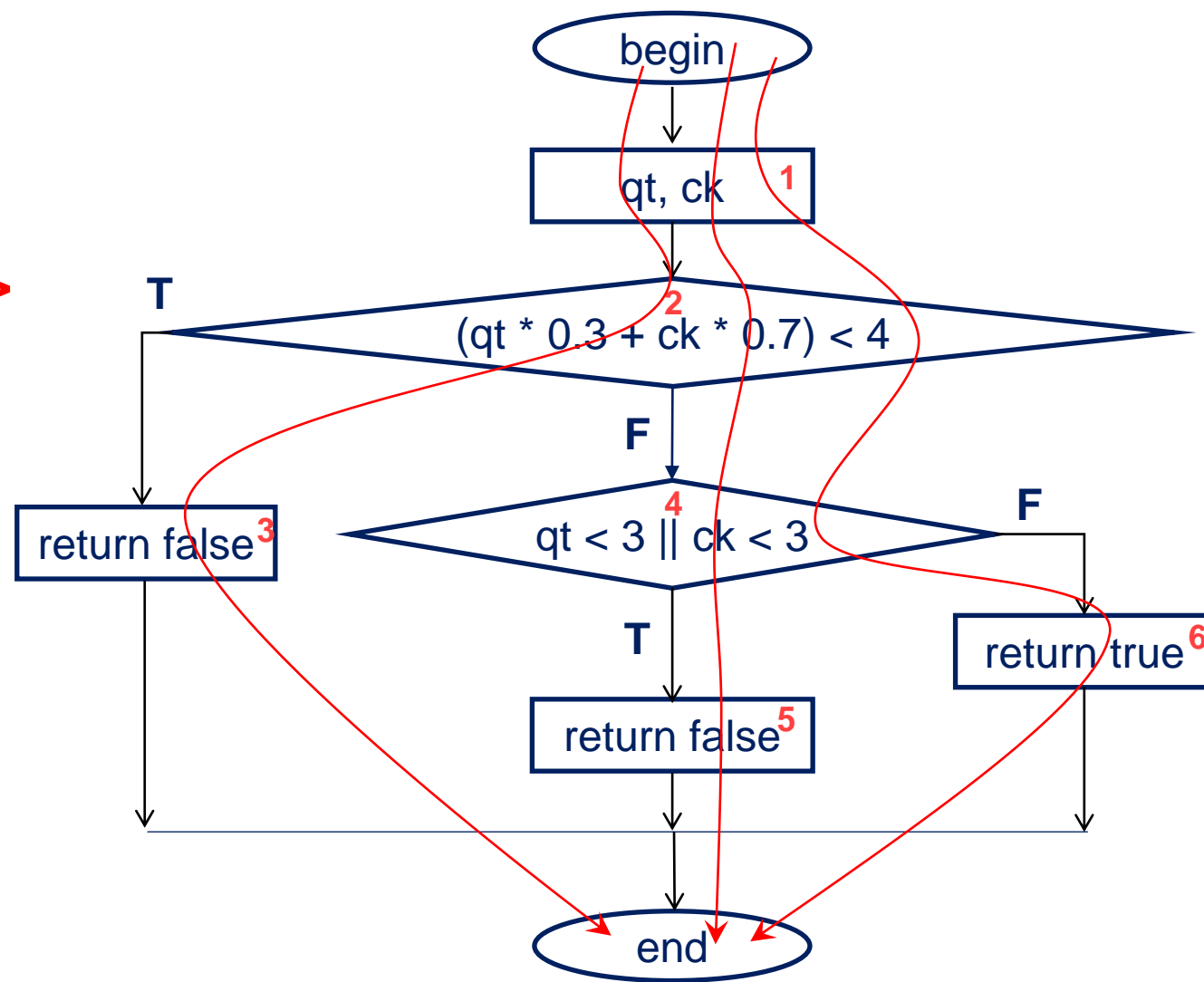


Hình 2.6: Sơ đồ luồng điều khiển – ví dụ 2

## 2. BÀI TẬP ÁP DỤNG

### Phủ đường dẫn

- (1 – 2 – 3): **TC <1, 5, false>**
- (1 – 2 – 4 – 5): **TC <2, 8, false>**
- (1 – 2 – 4 – 6): **TC <4, 8, true>**



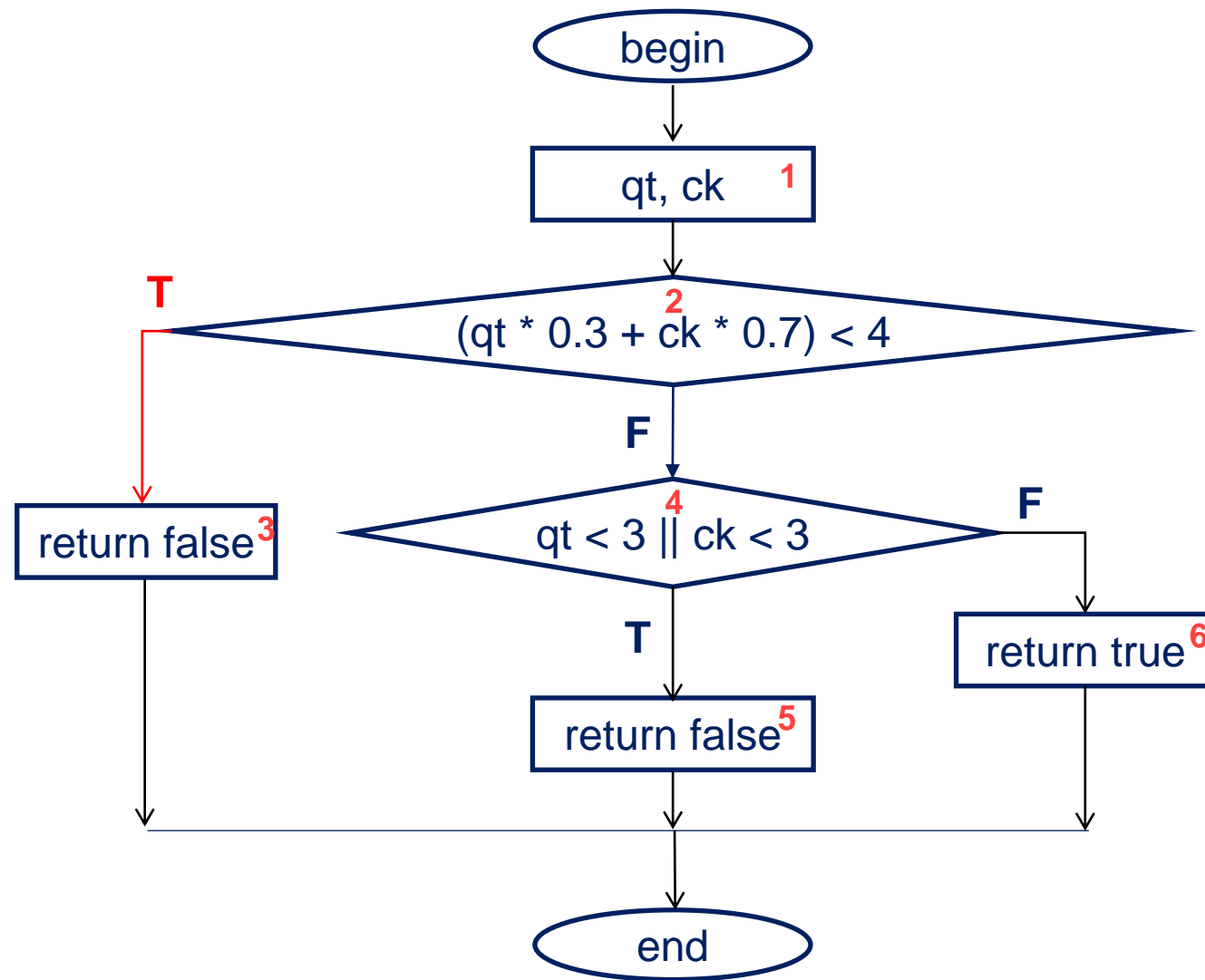
Hình 2.7: Sơ đồ luồng điều khiển – ví dụ 2

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ nhánh (1): **T**

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + ckT * 0.7) < 4)  
        return false;  
    else  
        if (qt < 3 || ck < 3)  
            return false;  
        else  
            return true;  
}
```

**TC <1, 5, false>**



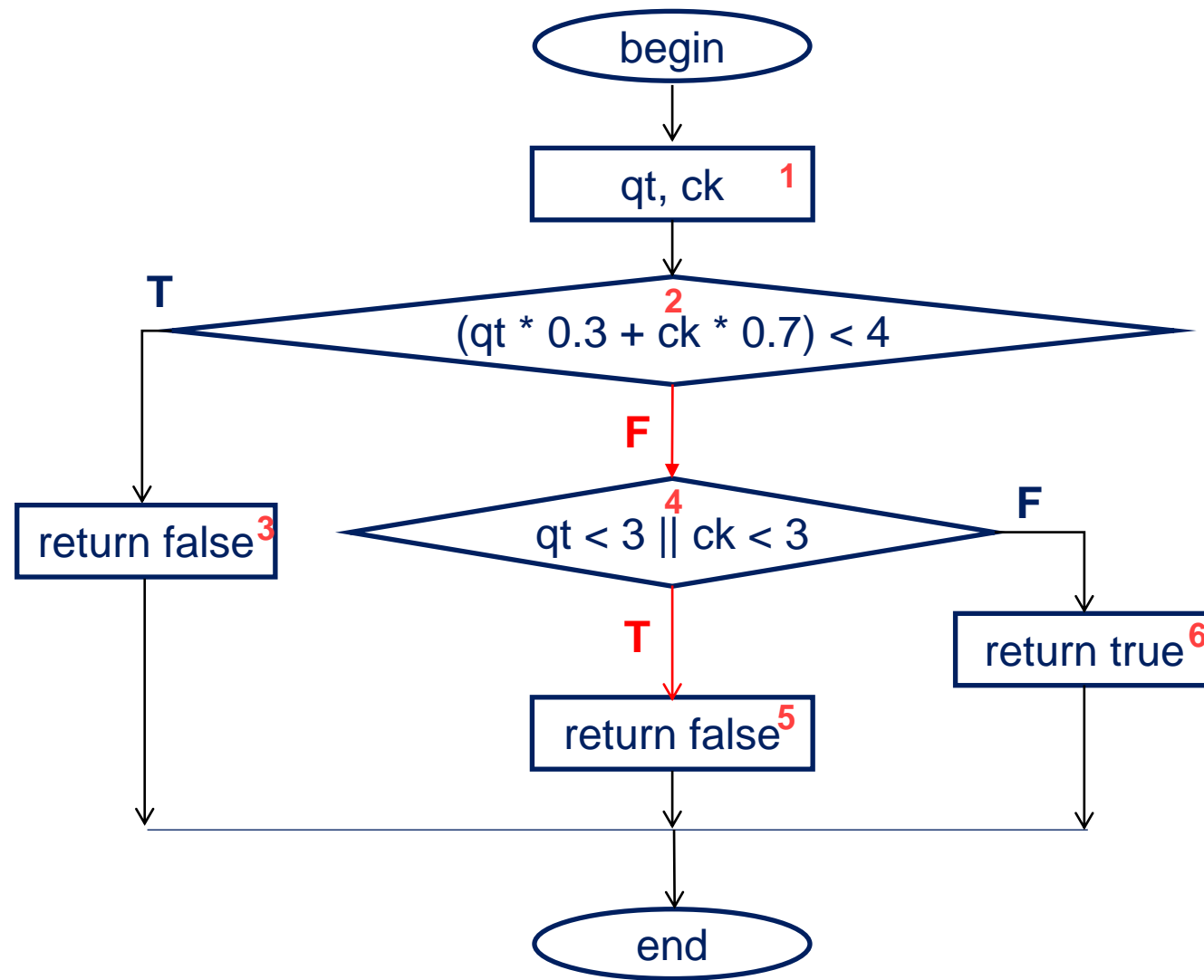
Hình 2.8:  $(qt * 0.3 + ck * 0.7) < 4$  (T)

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ nhánh (2): F, T

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + ckF * 0.7) < 4)  
        return false;  
    else  
        if (qt < 3T || ck < 3)  
            return false;  
        else  
            return true;  
}
```

**TC <2, 8, false>**



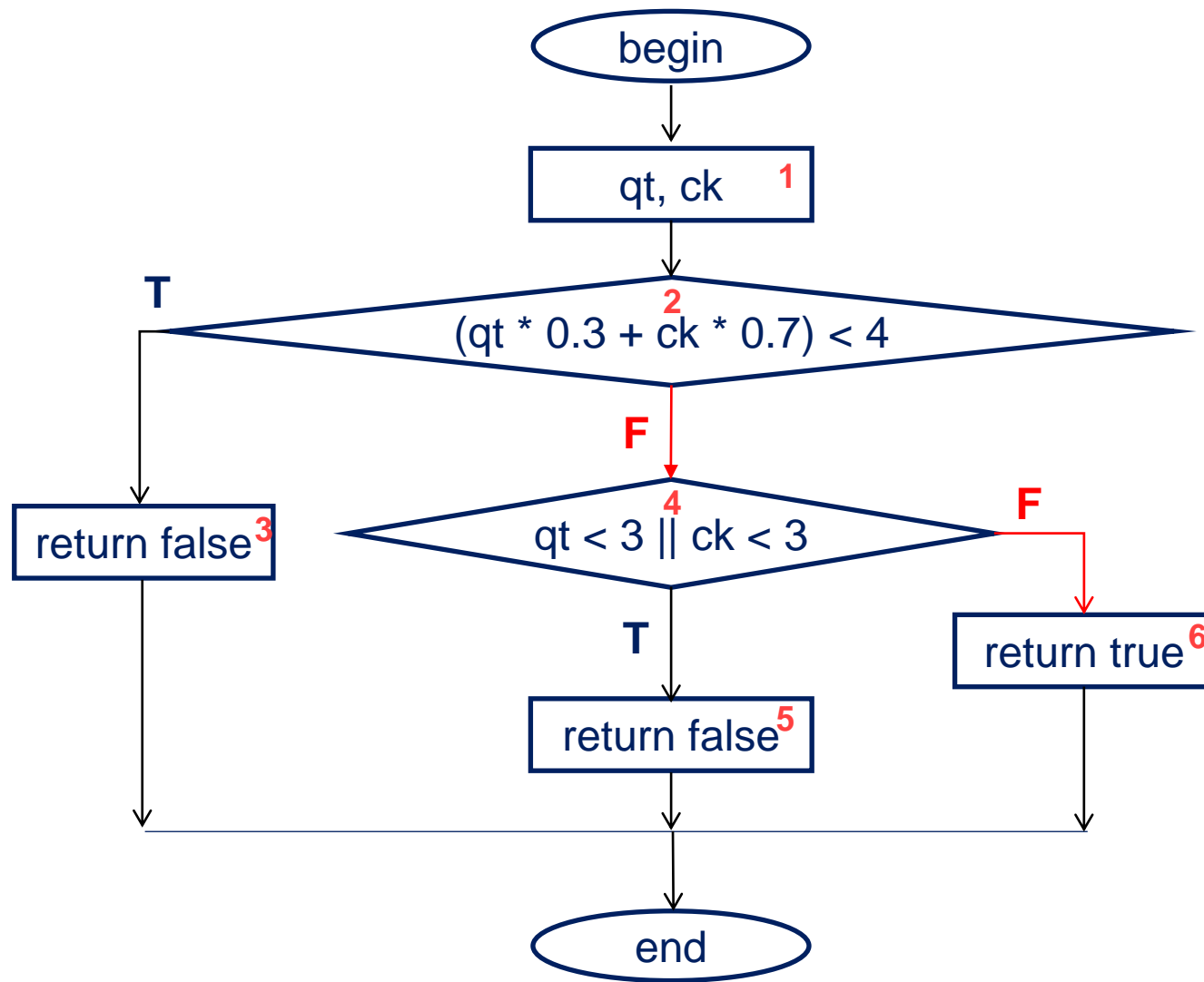
Hình 2.9:  $(qt * 0.3 + ck * 0.7) < 4$  (F) và  $(qt < 3 \parallel ck < 3)$  (T)

## 2. BÀI TẬP ÁP DỤNG

### Bao phủ nhánh (3): F, **F**

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + Fck * 0.7) < 4)  
        return false;  
    else  
        if (qt < 3 F || ck < 3)  
            return false;  
        else  
            return true;  
}
```

**TC <4, 8, true>**



Hình 2.10:  $(qt * 0.3 + ck * 0.7) < 4$  (F) và  $(qt < 3 \parallel ck < 3)$  (F)

## 2. BÀI TẬP ÁP DỤNG

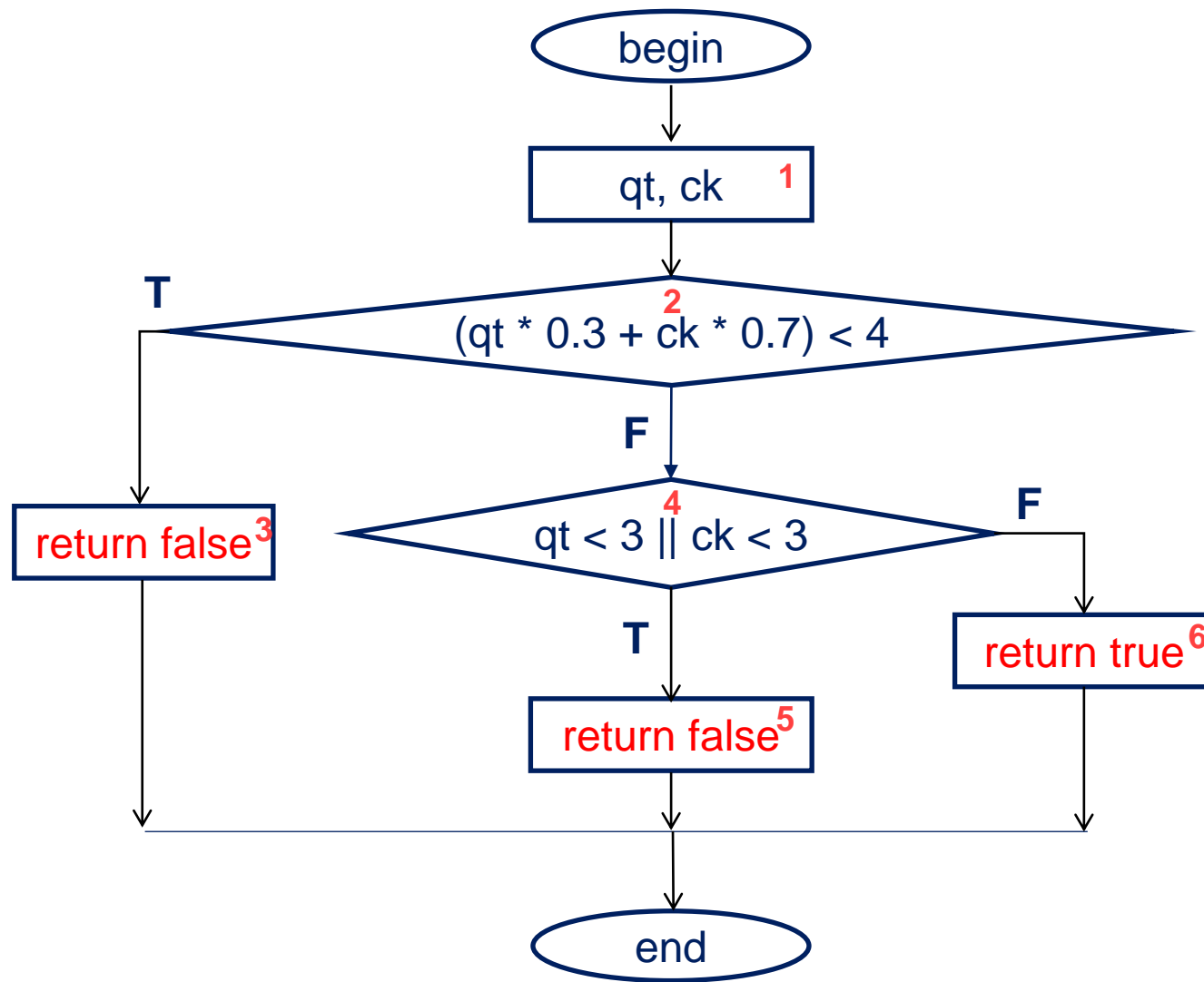
### Bao phủ lệnh

```
bool kq(float qt, float ck) {  
    if ((qt * 0.3 + ck * 0.7) < 4)  
        return false; //1  
    else  
        if (qt < 3 || ck < 3)  
            return false; //2  
        else  
            return true; //3  
}
```

//1: TC <1, 5, false>

//2: TC <2, 8, false>

//3: TC <4, 8, true>



Hình 2.11: Bao phủ lệnh

## 2. BÀI TẬP ÁP DỤNG

---



### Nhận xét

- Trong bài tập này cả 3 test case tìm được đều áp dụng với cùng 3 tiêu chí đã cho.
  1. TC <1, 5, false>
  2. TC <2, 8, false>
  3. TC <4, 8, true>
- Như vậy không có sự phân biệt rõ ràng tiêu chí nào là mạnh nhất hay yếu nhất.



1. Bài tập vận dụng ở trên đã cung cấp cho người học cách thức giải quyết bài toán kiểm thử theo đồ thị luồng điều khiển.
2. Tiếp sau bài này, **người học có thể tự tìm hiểu thêm** về các kĩ thuật kiểm thử hộp đen.

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

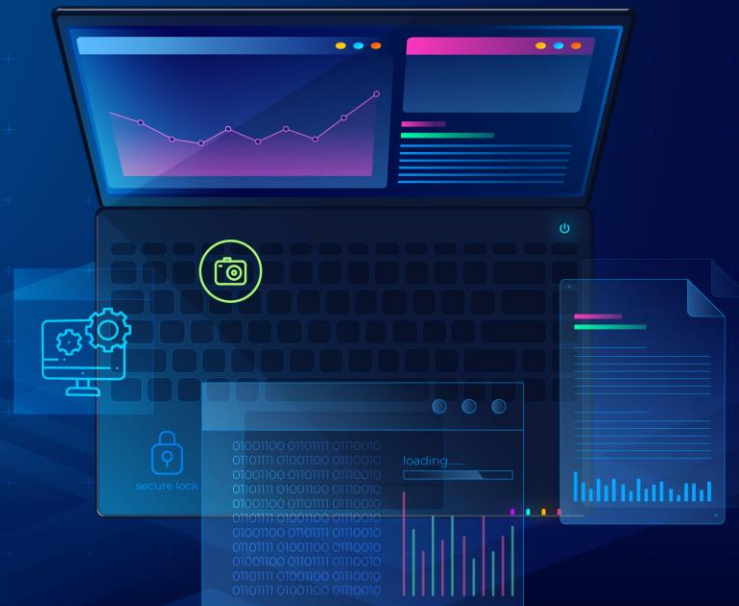
## Hướng dẫn bài tập: Phương pháp kiểm thử hộp trắng

Biên soạn:

TS. Trần Nhật Hóa

Trình bày:

TS. Trần Nhật Hóa



# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

*Bài học tiếp theo:*

## **Phương pháp kiểm thử hộp đen**

**Tài liệu tham khảo:**

- [1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016.
- [2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.
- [3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.
- [4] Shari Lawrence Pleege, Joanne M. Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009