

```
#include <Servo.h>

Servo doorServo1;
Servo doorServo2;
Servo doorServo3;

char currentFace = '0';

void setup() {
    Serial.begin(9600);
    doorServo1.attach(9);
    doorServo2.attach(10);
    doorServo3.attach(11);

    doorServo1.write(0);
    doorServo2.write(0);
    doorServo3.write(0);

    Serial.println("시스템 시작: 모든 문 열림");
}

void loop() {
    if (Serial.available() > 0) {
        char face = Serial.read();

        // 얼굴 인식 결과 변경되었을 때만 처리
        if (face != currentFace) {
            currentFace = face;

            // 모든 문 열기
            doorServo1.write(0);
            doorServo2.write(0);
            doorServo3.write(0);

            // 현재 face 값 출력
            Serial.print("인식된 혼잡도(face): ");
            Serial.println(currentFace);

            // 해당 서보만 열기
            if (face == '1') {
                doorServo1.write(80);
                Serial.println("1번 길 닫힘 (서보1 90도)");
                delay(3000);
            }
        }
    }
}
```

```

        doorServo1.write(0);
    } else if (face == '2') {
        doorServo2.write(80);
        Serial.println("2번 길 닫힘 (서보2 90도)");
        delay(3000);
        doorServo2.write(0);
    } else if (face == '3') {
        doorServo3.write(80);
        Serial.println("3번 길 닫힘 (서보3 90도)");
        delay(3000);
        doorServo3.write(0);
    } else {
        Serial.println("알 수 없는 face 값. 모든 문 닫힘.");
    }
}
}

// 현재 서보 상태 주기적으로 출력 (디버깅 용도)
delay(1000);
Serial.print("현재 상태 - face: ");
Serial.print(currentFace);
Serial.print(" | 서보1: ");
Serial.print(doorServo1.read());
Serial.print("°, 서보2: ");
Serial.print(doorServo2.read());
Serial.print("°, 서보3: ");
Serial.print(doorServo3.read());
Serial.println("°");
}

```