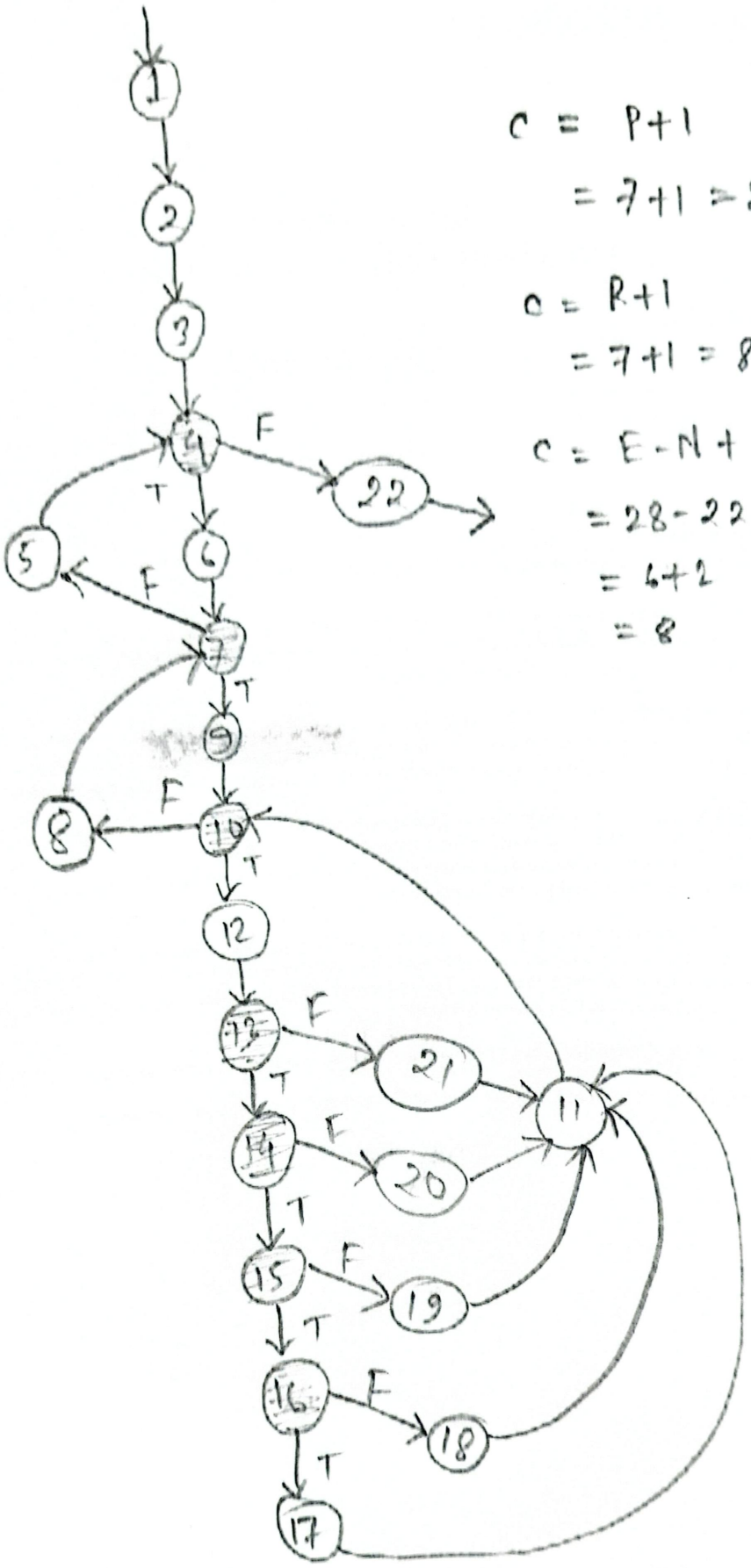
1. Draw the **CFG** (mark the nodes on the question paper), find the **cyclomatic complexity** using **three** formulae. Write **three paths** from the basic path set. **[5+3+2]**

```
def complex_function(matrix): ===> node 1 (start the node count from here)
    result = []  -> 2
    for i in range(0, len(matrix), 1):
        for j in range(0, len(matrix[i]), 1):
            for k in range(0, len(matrix[i][j]), 1):
                val = matrix[i][j][k]
                if val % 2 == 0:
                    if val > 50:
                        if val % 5 == 0:
                            if val < 100:
                                result.append("Special Even")
                            else:
                                result.append("Large Even")
                        else:
                            result.append("Even > 50")
                    else:
                        result.append("Small Even")
                else:
                    result.append("Odd")
    return result
```



$$c = P + 1$$
$$= 7 + 1 = 8$$

$$c = R + 1$$
$$= 7 + 1 = 8$$

$$c = E - N + 2P$$
$$= 28 - 22 + 2$$
$$= 6 + 2$$
$$= 8$$

1. Draw the **CFG** (mark the nodes on the question paper), find the **cyclomatic complexity** using **three** formulae. Write **three paths** from the basic path set. **[5+3+2]**

```
def cooling_system_simulation(): # this is -- node 2
    temperature = 50
    max_temperature = 80
    fan_speed = 5
    cooling_rate = 2
    is_cooling_system_on = False
    is_emergency_mode = False

    while temperature > 0:
        if is_cooling_system_on:
            for i in range(0, fan_speed, 1):
                if temperature < max_temperature:
                    for j in range(0, cooling_rate, 1):
                        temperature -= 1
                        print(f"Cooling... Current temperature: {temperature}")
                else:
                    print("Temperature is too high!")
        else:
            is_cooling_system_on = False
            temperature += 1
            print(f"Warming... Current temperature: {temperature}")

        if is_emergency_mode:
            temperature -= 5
            return "Emergency mode activated! Forcing rapid cooling."

        if temperature <= 0:
            print("System shutting down. Temperature is safe.")
            break

if __name__ == "__main__":
    print(cooling_system_simulation()) # start from here -- node 1
```

$c = P+1 = 7+1 = 8$

$c = R+1 = 7+1 = 8$

$c = E-N+2P = 28-22+2\times1$
$= 8$