

## CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h> // For sleep()

// Simulated sensor readings
float readTemperature() { return 25.0; }
float readTDS() { return 300.0; } //ppm
float readPH() { return 7.0; } //Neutral pH
bool solarChargingAvailable() { return true; }
bool usbChargingAvailable() { return true; }
bool isBatteryCharged() { return true; }

// Check if water is suitable for UV purification
bool isWaterSafe(float tds, float ph) {
    return (tds >= 50 && tds <= 500) && (ph >= 6.5 && ph <= 8.5);
}

void startPurification() {
    printf("UV-C LED activated. Purification in progress...\n");
    for (int i = 0; i < 3; i++) {
        printf("Purifying (%d/3)...\n", i+1);
        sleep(1);
    }

    printf("Purification complete. UV-C LED off.\n");
}

void automaticShutOff() {
    printf("Auto shut-off triggered. Device is now in standby mode.\n");
}

void checkCharging() {
    if (solarChargingAvailable())
        printf("Charging via solar panel...\n");
    else if (usbChargingAvailable())
        printf("Charging via USB-C...\n");
    else
        printf("No charging source available.\n");
}
```

```

int main() {
    float temp =
    readTemperature();
    float tds =
    readTDS();
    float ph = readPH();

    printf("Temp: %.2f°C | TDS: %.2f ppm | pH: %.2f\n", temp, tds, ph);
    checkCharging();

    if (!isBatteryCharged()) {
        printf("Battery low! Cannot proceed with
        purification.\n"); return 1;
    }

    if (isWaterSafe(tds, ph)) {
        printf("Water within safe
        parameters.\n");
        startPurification();
    } else {
        printf("Water quality poor! Not suitable for UV purification.\n");
    }

    automati
    cShutOff
    (); return
    0;
}

```