```python
#「目標温度をセンサ｛サーミスタ｝の値に応じて出力｛ヒーター｝を
#操作することによって達成する」（＝制御）プログラム

import spidev
import time
import numpy as np
import math

import RPi.GPIO as GPIO

target_temperature = 50
Kp = 20
Ki = 0.5

# SPI通信の設定
spi = spidev.SpiDev()
spi.open(0, 0)                          #SPI"0"のCS"0"を利用

# 出力ピンの設定
PIN_HEATER = xx
GPIO.setmode(GPIO.BCM)
GPIO.setup(PIN_HEATER,GPIO.OUT)         #ピンに出力（入力ではない）を設定
pwm =  GPIO.PWM(PIN_HEATER,0)           #ピンにPWMを設定

# 時間とデータを保存するためのNumPy配列の作成
timestamps = np.array([])
temperatures = np.array([])

time_start = time.time()                #プログラムの開始時間の取得
pwm.start()                             #PWM出力の開始

try:
    print("ADコンバータのデータを記録中...Ctrl+Cで終了します。")
    while True:#ループの開始

        #データの取得
        resp = spi.xfer2([0x68, 0x00])
                                        # SPI通信でADCからデータを取得
        adc_value = ((resp[0] << 8) + resp[1]) & 0x3FF
                                        # 読んだ値を0-1023の数値に変換
        timestamp = time_start - time.time()
                                        #「現在の時間」を計算
        print(f"Time: {timestamp}, Value: {adc_value}")
                                        #データを表示

        #得られたvalueから，温度を計算（8-1）
        voltage = value/1023*3.3        #ADC取得値を電圧に変換
        r_therm = (3.3-voltage)/voltage*10000
                                        #サーミスタの抵抗値を算出
                                        #(10000はサーミスタと直列な抵抗値)
        temperature = 1/(1/(25+273.15)+1/3435*math.log(r_therm/10000))
                                        #B定数の式でサーミスタの温度を変換
```

```
            temperature = temperature - 273.15
                                        #求めた絶対温度をセ氏温度に変換
            print(f"Temperature: {temperature}度")

            #得られた温度から出力を計算 (9-1,2)
            """
            #①バンバン制御
            if temperature > target_temperature:
                #GPIO.output(PIN_HEATER,GPIO.HIGH)
                output = 100            #最大出力（オン）
            else:
                #GPIO.output(PIN_HEATER,GPIO.LOW)
                output = 0              #最小出力（オフ）
            """


            """
            #②P制御
            error = target_temperature - temperature
            output = error * Kp         #P制御の計算
            """
            """
            #③PI制御
            error = target_temperature - temperature
            errorsum = errorsum + error
            output = error * Kp + error_sum * Ki
                                        #PI制御の計算
            """


            #計算した出力をPWMの範囲に変換
            if output > 100:
                output = 100
            elif output <= 0:
                output = 0
            pwm.ChangeDutyCycle(output)     #計算した出力をPWMとして出力

            #データ(温度・時間)を，保存のための配列に追加
            timestamps = np.append(timestamps, timestamp)
            temperatures = np.append(temperatures, temperature)

            time.sleep(0.1)                 #ループ時間の設定：0.1秒


    except KeyboardInterrupt:               #ループを抜け，記録終了後の処理
        print("¥nデータ記録を終了します。")
        print("記録された時間:")
        print(timestamps)
        print("記録されたデータ:")
        print(values)

        ###データの保存
        data = np.vstack((timestamps, values))
                                        #時間とデータの配列を縦に結合
```

```
105         filename = f"adc_data_{time.strftime('%Y%m%d_%H%M%S')}.csv"
106                                                 # CSVファイル名(string)の作成
107
108         np.savetxt(                             #CSVに保存
109             filename,                           #保存する先のファイル名
110             data,                               #保存する行列data
111             delimiter=",",                      #CSV形式で配列を格納
112             fmt="%.6f",                         #数字を下6桁までにする
113             header="Timestamps (row 1),Values (row 2)",
114                                                 # ヘッダーの追加
115             comments=""                         # ヘッダー行に#を付けない
116         )
117         print(f"データを{filename}に保存しました。")
118
119         input("Enterキーを押して終了してください...")
120                                                 #終了前にターミナルを待機
121
122
123     finally:
124         spi.close()                             #spi通信の終了
125
126
```