(Question 1)

Assume a microprocessor-controlled room heating management system, comprising of a heater and a thermometer. Write an algorithm, pseudocode to maintain the room temperature at constant 25°C.

(Algorithm/Pseudocode)

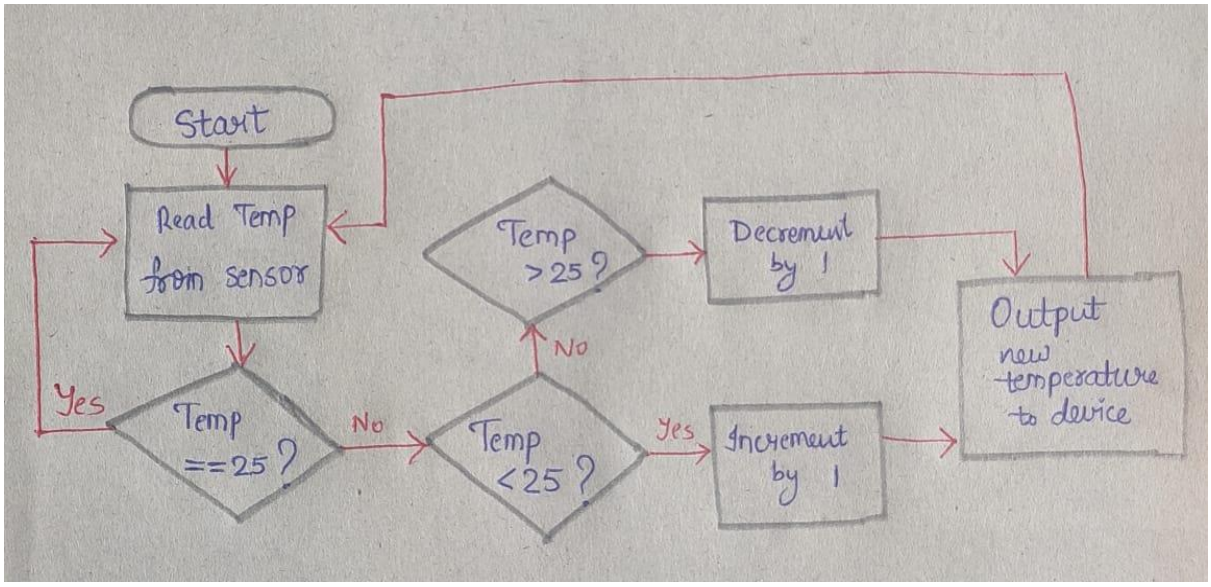
START

Read temperature from sensor
Compare with 25
IF equal, loop back to start

IF less, jump to below
ELSE, decrement temperature by 1
Output new temperature value to device
Jump to start
IN below, increment temperature by 1
Output new temperature value to device
Jump to start

END

(Flowchart)



(Question 2)

Develop an ALP to maintain the temperature at 25°C using IN and OUT commands.

(Aim)

To maintain the temperature at constant of 25°C.

(ALP Code)

```
.model small
.stack 100h
.data
   temp db 25
.code
   mov ax,@data
   mov ds,ax

start:
   IN al,01H        ; read temperature value from sensor connected to port 01H
   cmp al,temp   ; compare the temperature value with the desired temperature (25°C)
   je start          ; jump back to start if the temperature is already at 25°C
   jc below         ; jump to below if the temperature is below 25°C

above:
   dec al          ; decrease the temperature by 1°C
   OUT 02H,al  ; output the new temperature value to a device connected to port 02H
   jmp start      ; jump back to start to read the temperature value again

below:
   inc al          ; increase the temperature by 1°C
   OUT 02H,al  ; output the new temperature value to a device connected to port 02H
   jmp start     ; jump back to start to read the temperature value again

end:
   mov Ax,4C00H   ; return control to operating system
   INT 21H
   END
```

(Result)

Branching instructions have been successfully implemented to maintain the temperature **at constant 25°C** using IN and OUT instructions and **incrementing or decrementing the temperature gradually till condition not met**.

(Question 3)

Develop an ALP to maintain temperature between 30°C to 40°C using IN and OUT commands.

(Aim)

To maintain the temperature in between the range of 30-40°C.

(Algorithm/Pseudocode)

START

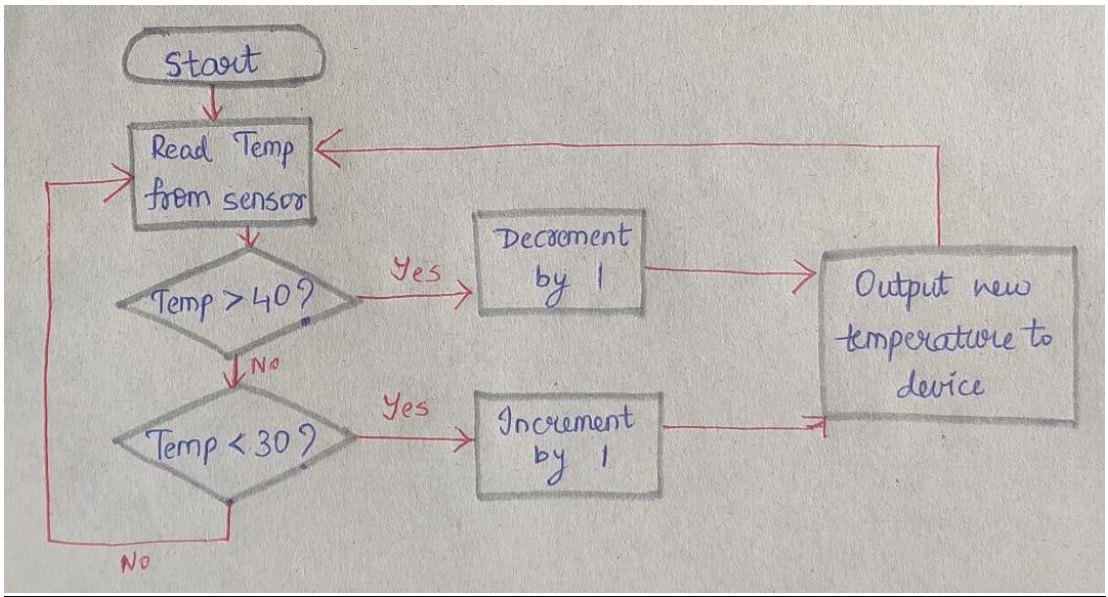
Read temperature from sensor
Compare with max_temp (40)
IF more, jump to above
Compare with min_temp (30)
IF less, jump to below
IF none, jump back to start

IN above, decrement temperature by 1
Output the temperature to the device
Jump to start

IN below, increment temperature by 1
Output the temperature to the device
Jump to start

END

(Flowchart)



(ALP Code)

```
.model small
.stack 100h
.data
   max_temp db 40
   min_temp db 30
.code
   mov ax,@data
   mov ds,ax

start:
   IN al,01H              ; read temperature value from sensor connected to port 01H
   cmp al,max_temp    ; compare the temperature with the maximum temperature
   jg above               ; jump to above label if the temperature is above 40°C
   cmp al,min_temp    ; compare the temperature with the minimum temperature
   JL below               ; jump to below label if the temperature is below 30°C
   jmp start              ; jump to start label if the temperature is within the desired range

above:
   dec al             ; decrease the temperature by 1°C
   OUT 02H,al         ; output the new temperature value to a device connected to port 02H
   jmp start          ; jump back to start label to read the temperature value again

below:
   inc al             ; increase the temperature by 1°C
   OUT 02H,al         ; output the new temperature value to a device connected to port 02H
   jmp start          ; jump back to start label to read the temperature value again

end:
   mov ax,4C00H       ; return control to operating system
   INT 21H
   END
```

(Result)

Branching instructions have been successfully implemented to maintain the temperature **within the range** using IN and OUT instructions and **incrementing or decrementing the temperature gradually till condition not met**.