Question:

Create a program that records and analyzes the vital signs of patients over a specific time period.

Requirements:

Input an array representing the heart rates of a patient recorded every minute for 24 hours (1440 readings).

Identify the following:

The average heart rate for the day.

The highest and lowest heart rate along with the time at which they occurred.

Any period of continuous heart rate below a specified threshold (e.g., 60 bpm) lasting more than 5 minutes.

Allow the user to input a range of indices and display the heart rate trend (values) for that period.

Constraints:

Validate that all heart rate readings are within the humanly possible range (30 bpm to 200 bpm).

Handle edge cases, such as all values being constant or invalid input ranges.

Code:-

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define TOTAL_READINGS 1440

// Function declarations
float calculate_average(int heart_rates[], int size);
void find_highest_lowest(int heart_rates[], int size, int *highest, int *lowest, int *high_time, int *low_time);
void find_low_threshold_periods(int heart_rates[], int size, int threshold);
void display_trend(int heart_rates[], int start, int end);
bool validate_heart_rates(int heart_rates[], int size);

int main() {
    int heart_rates[TOTAL_READINGS];
    int highest, lowest, high_time, low_time;
    int threshold = 60;
```

```
// Input heart rates
printf("Enter %d heart rate readings (1 per minute):\n", TOTAL_READINGS);
for (int i = 0; i < TOTAL_READINGS; i++) {
   scanf("%d", &heart_rates[i]);
}

// Validate heart rates
if (!validate_heart_rates(heart_rates, TOTAL_READINGS)) {
   printf("Error: Heart rate readings must be between 30 and 200 bpm.\n");
   return 1;
}

// Calculate average heart rate
float average = calculate_average(heart_rates, TOTAL_READINGS);

// Find highest and lowest heart rates
find_highest_lowest(heart_rates, TOTAL_READINGS, &highest, &lowest, &high_time, &low_time);

// Print results
printf("\nAverage heart rate: %.2f bpm\n", average);
printf("Highest heart rate: %d bpm at minute %d\n", highest, high_time);
printf("Lowest heart rate: %d bpm at minute %d\n", lowest, low_time);

// Identify low-threshold periods
printf("\nContinuous periods with heart rate below %d bpm:\n", threshold);
find_low_threshold_periods(heart_rates, TOTAL_READINGS, threshold);

// Allow user to display trend for a range
```

```
    int start, end;
    printf("\nEnter a range of indices (start and end) to display the heart rate trend: ");
    scanf("%d %d", &start, &end);

    if (start >= 0 && end < TOTAL_READINGS && start <= end) {
        printf("\nHeart rate trend from minute %d to %d:\n", start, end);
        display_trend(heart_rates, start, end);
    } else {
        printf("Invalid range. Please ensure 0 <= start <= end < %d.\n", TOTAL_READINGS);
    }

    return 0;
}

// Function to calculate the average heart rate
float calculate_average(int heart_rates[], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += heart_rates[i];
    }
    return (float)sum / size;
}

// Function to find the highest and lowest heart rates
void find_highest_lowest(int heart_rates[], int size, int *highest, int *lowest, int *high_time, int *low_time) {
    *highest = heart_rates[0];
    *lowest = heart_rates[0];
    *high_time = 0;
    *low_time = 0;
```

```
    for (int i = 1; i < size; i++) {
        if (heart_rates[i] > *highest) {
            *highest = heart_rates[i];
            *high_time = i;
        }
        if (heart_rates[i] < *lowest) {
            *lowest = heart_rates[i];
            *low_time = i;
        }
    }
}

// Function to identify continuous low-threshold periods
void find_low_threshold_periods(int heart_rates[], int size, int threshold) {
    int start = -1, count = 0;

    for (int i = 0; i < size; i++) {
        if (heart_rates[i] < threshold) {
            if (start == -1) {
                start = i;
            }
            count++;
        } else {
            if (count > 5) {
                printf("Start: minute %d, End: minute %d\n", start, start + count - 1);
            }
            start = -1;
            count = 0;
```

```
        }
    }

    if (count > 5) {
        printf("Start: minute %d, End: minute %d\n", start, start + count - 1);
    }
}

// Function to display heart rate trend in a given range
void display_trend(int heart_rates[], int start, int end) {
    for (int i = start; i <= end; i++) {
        printf("Minute %d: %d bpm\n", i, heart_rates[i]);
    }
}

// Function to validate heart rate readings
bool validate_heart_rates(int heart_rates[], int size) {
    for (int i = 0; i < size; i++) {
        if (heart_rates[i] < 30 || heart_rates[i] > 200) {
            return false;
        }
    }
    return true;
}
```