Signal Processing with Threshold Marking

You are tasked with extending the signal processing project to include a threshold marking mechanism. Your goal is to:

Define a marker value (SIGPROCMARK) to mark the processed signal elements.

Implement a function processSignalWithThreshold that processes each element of the signal by doubling its value only if it is greater than a given threshold, and then marking it with SIGPROCMARK.

Implement a function displaySignal to print the signal values to the console.

Demonstrate the usage of these functions in a main function with an example signal and a threshold value.

Requirements:

The marker value should be defined as a constant.

The processSignalWithThreshold function should double the value of each element that exceeds the threshold and use bitwise operations to mark the processed elements.

The displaySignal function should print the signal values separated by spaces.

#include <iostream>

#include <vector>

// Define the marker value

const int SIGPROCMARK = 0x80000000;

// Function to process the signal with a given threshold

void processSignalWithThreshold(std::vector<int>& signal, int threshold) {

for (int& element : signal) {

if (element > threshold) {

element = (element \* 2) | SIGPROCMARK;

}

}

}

// Function to display the signal values

void displaySignal(const std::vector<int>& signal) {

for (int element : signal) {

std::cout << element << " ";

}

std::cout << std::endl;

}

// Main function to demonstrate the usage

int main() {

// Example signal and threshold value

std::vector<int> signal = {5, 15, 25, 35, 10};

int threshold = 20;

// Process the signal with the given threshold

processSignalWithThreshold(signal, threshold);

// Display the processed signal

displaySignal(signal);

return 0;

}