```cpp
#include "ece420_main.h"
#include "ece420_lib.h"
#include "kiss_fft/kiss_fft.h"

// JNI Function
extern "C" {
JNIEXPORT float JNICALL
Java_com_ece420_lab4_MainActivity_getFreqUpdate(JNIEnv *env, jclass);
}

// Student Variables
#define F_S 48000
#define FRAME_SIZE 1024
#define VOICED_THRESHOLD (1800000000/2048)*FRAME_SIZE  // Find your own threshold
float lastFreqDetected = -1;


void ece420ProcessFrame(sample_buf *dataBuf) {
    // Keep in mind, we only have 20ms to process each buffer!
    struct timeval start;
    struct timeval end;
    gettimeofday(&start, NULL);

    // Data is encoded in signed PCM-16, little-endian, mono
    float bufferIn[FRAME_SIZE];
    for (int i = 0; i < FRAME_SIZE; i++) {
        int16_t val = ((uint16_t) dataBuf->buf_[2 * i]) | (((uint16_t) dataBuf->buf_[2 * i + 1]) << 8);
        bufferIn[i] = (float) val;
    }
```

```
    // write your determined frequency. If unvoiced, write -1.
    // ******************** START YOUR CODE HERE ********************** //
    //float threshold = (1800000000/2048)*FRAME_SIZE;


    float E = 0;
    for (int i = 0; i<FRAME_SIZE; i++){
        E += bufferIn[i] * bufferIn[i];
    }

    if (E < VOICED_THRESHOLD){
        lastFreqDetected = -1;

    }
    else {

        kiss_fft_cpx fin[FRAME_SIZE];
        kiss_fft_cpx fout[FRAME_SIZE];
        kiss_fft_cpx fmulti[FRAME_SIZE];
        kiss_fft_cpx output[FRAME_SIZE];

        for (int k = 0; k < FRAME_SIZE; k++) {
            fin[k].r = bufferIn[k];
            fin[k].i = 0;
        }
        kiss_fft_cfg cfg = kiss_fft_alloc(FRAME_SIZE, 0, NULL, NULL);
        kiss_fft(cfg, fin, fout);
        for (int j = 0; j < FRAME_SIZE; j++) {
            fmulti[j].r = fout[j].r * fout[j].r - (-1 * fout[j].i) * fout[j].i;
            fmulti[j].i = fout[j].r * fout[j].i + (-1 * fout[j].i) * fout[j].r;
        }
        kiss_fft_cfg cfg_ = kiss_fft_alloc(FRAME_SIZE, 1, NULL, NULL);
        kiss_fft(cfg_, fmulti, output);

        float stuff[FRAME_SIZE];
        for (int itr = 0; itr < FRAME_SIZE; itr++) {
            stuff[itr] = output[itr].r;
        }

        int maxIdx = findMaxArrayIdx(stuff, int(F_S / 270), int(F_S / 60));

        lastFreqDetected = F_S / maxIdx;
    }

    // ******************** END YOUR CODE HERE ********************** //
    gettimeofday(&end, NULL);
```