```
bool lab5PitchShift(float *bufferIn) {
    // Lab 4 code is condensed into this function
    int periodLen = detectBufferPeriod(bufferIn);
    float freq = ((float) F_S) / periodLen;

    // If voiced
    if (periodLen > 0) {

        LOGD("Frequency detected: %f\r\n", freq);

        // Epoch detection - this code is written for you, but the principles will be quizzed
        std::vector<int> epochLocations;
        findEpochLocations(epochLocations, bufferIn, periodLen);

        // In this section, you will implement the algorithm given in:
        // https://courses.engr.illinois.edu/ece420/lab5/lab/#buffer-manipulation-algorithm
        //
        // Don't forget about the following functions! API given on the course page.
        //
        // getHanningCoef();
        // findClosestInVector();
        // overlapAndAdd();
        // ********************** START YOUR CODE HERE  ************************** //
        int new_epoch_spacing = F_S/FREQ_NEW;
        int epoch_mark = 0;

        //removing duplicates in epochlocations
        //auto target = std::unique(epochLocations.begin(),epochLocations.end());
        //epochLocations.erase(target);

        //getting new epoch locations
        while (newEpochIdx < FRAME_SIZE*2){
        //for (newEpochIdx; newEpochIdx<FRAME_SIZE*2; newEpochIdx+=new_epoch_spacing){
            auto itr = findClosestInVector(epochLocations,newEpochIdx,epoch_mark,(epochLocations).size()-1);
            epoch_mark = itr;

            auto p0 = abs(epochLocations[itr-1] - epochLocations[itr+1])/2;

            //window generation
            std::vector<float> window((int)p0*2);
            for (int y=0; y<2*p0+1;y++){
                window[y] = getHanningCoef(p0*2,y);
            }
```

```
76                //window application
77                std::vector<float> windowed_sample(2*(int)p0+1);
78                for (int z=0;z<2*p0;z++){
79                    int ptr = epochLocations[itr]-p0+z;
80                    windowed_sample[z] = window[z]*bufferIn[ptr];
81                }
82
83                //sample localization
84                overlapAddArray(bufferOut,windowed_sample.data(),newEpochIdx-p0,2*p0);
85                newEpochIdx+=new_epoch_spacing;
86            }
87
88
89
90
91            // ********************** END YOUR CODE HERE  *************************** //
92        }
93
94        // Final bookkeeping, move your new pointer back, because you'll be
95        // shifting everything back now in your circular buffer
96        newEpochIdx -= FRAME_SIZE;
97        if (newEpochIdx < FRAME_SIZE) {
98            newEpochIdx = FRAME_SIZE;
99        }
100
101       return (periodLen > 0);
102   }
103
```