Digital Signal Processing (MGK)

(a) Consider a digital filter with impulse response

\[ h_i = 2\alpha \cdot \frac{\sin[2\pi(i - n/2)\alpha]}{2\pi(i - n/2)\alpha} \cdot w_i \]

where \( w_i = \begin{cases} 1, & 0 \leq i \leq n \\ 0, & \text{otherwise} \end{cases} \).

(i) What type of filter is this? [4 marks]

(ii) How are the sampling rate \( f_s \) at which this filter is operated and its \(-6\) dB cut-off frequency \( f_c \) related to parameter \( \alpha \)? [2 marks]

(b) In an open-source audio-effect library, you find a C routine for processing a recorded voice to sound like it came over an analog phone line:

```c
#include <math.h>
#define N 512
#define PI 3.14159265358979323846
void phone_effect(double *x, double *y, int m)
{
    double w, p, f, g, h[N+1];
    int i, k;
    for (i = 0; i <= N; i++) {
        w = 0.54 - 0.46 * cos(2*PI*i/N);
        p = 2 * PI * (i-N/2) / 10;
        f = w * ((p == 0) ? 1 : sin(p)/p) / 5;
        p = 2 * PI * (i-N/2) / 100;
        g = w * ((p == 0) ? 1 : sin(p)/p) / 50;
        h[i] = f - g;
    }
    for (i = 0; i < m; i++) {
        y[i] = 0;
        for (k = 0; k <= N && k <= i; k++)
            y[i] += x[i - k] * h[k];
    }
}
```

The input array \( x \) and the output array \( y \) each hold \( m \) samples of an audio recording (mono) at sampling frequency \( f_s = 32 \) kHz.

(i) Explain in detail what operation is implemented here (e.g., type of filter, order, cut-off frequency) and how it has been constructed. [8 marks]

(ii) You want to use this algorithm on audio recordings with a sampling rate of 48 kHz. What do you have to change in the source code to ensure that the audible effect remains the same? [6 marks]