Algorithm Convert Optimization Logs to Readable Format - P1 Converter

Input: File `solution.sol` with variables Sijk, Cijk, Xijk
Output: File `solution_readable.txt` with sorted operations by start and end times
Load file `solution.sol` into `lines`
Initialize dictionary `operation_times` to store Sijk, Cijk, Xijk values
function round_value(value)
    if —value— ¡ 0.001 **then**
        return 0
    else
        return **round**(value)
    end if
end function
for each line in `lines` **do**
    if line starts with "Sijk", "Cijk", or "Xijk" **then**
        Extract `var_name` and `value`
        Extract (i, j, k): Job, Operation, Resource
        if (i, j, k) not in `operation_times` **then**
            Initialize entry for (i, j, k) in `operation_times`
        end if
        if `var_name` contains "Sijk" **then**
            Store `value` in `operation_times[(i,j,k)]['start']`
        else if `var_name` contains "Cijk" **then**
            Store `value` in `operation_times[(i,j,k)]['end']`
        else if `var_name` contains "Xijk" and value ¿ 0.5 **then**
            Mark operation as active in `operation_times[(i,j,k)]['active']`
        end if
    end if
end for
Filter operations to remove inactive or zero-duration operations
Initialize `filtered_operations_final`
for each (i, j, k) in filtered operations **do**
    if a collaborative operation "Co" exists for this job **then**
        if (i, j, k) is the "Co" operation **then**
            Keep only this "Co" operation in `filtered_operations_final`
        end if
    else
        Add current operation to `filtered_operations_final`
    end if
end for
Sort `filtered_operations_final` by start time
Write sorted operations to `solution_readable.txt`
for each (i, j, k) and times in sorted operations **do**
    Write: "Operation {j}, Job {i}, Resource {k}: Start = {times['start']}, End = {times['end']}"
end for
return "Processing completed. Results in `solution_readable.txt`."