```julia
## IO.jl

function read_instance(filename::String)
    f = open(filename)
    readline(f)#comment
    n_jobs, n_processors, UB = parse.(Int,split(readline(f)))
    readline(f)#comment
    duration = zeros(Int,n_jobs,n_processors)
    for i in 1:n_jobs
        duration[i,:] = parse.(Int,split(readline(f)))
    end
    readline(f)#comment
    processor = zeros(Int,n_jobs,n_processors)
    for i in 1:n_jobs
        processor[i,:] = parse.(Int,split(readline(f)))
    end
    close(f)

    path, file = split(filename, ".")
    dir, name = split(path, "/")
    return n_jobs, # the number of jobs
           n_processors, # the number of processors = number of operations
           UB, # the best-known upper bound
           duration, # the duration of each operation
           processor, # the processor assinged to each operation
           name
end

function writeSolution(solution, solutionLocation, n_jobs, n_processors)
    wDir = string(pwd())

    dir, file = splitdir(solutionLocation)
    if (!isdir(dir))
        mkpath(string("./", dir, "/"))
    end

    open(string(wDir, "/", solutionLocation), "w") do f
        for i in 1:n_jobs
            for j in 1:n_processors
                for processor in solution
                    for job in processor
                        if (i == job[1] && j == job[2])
                            write(f, string(job[3], " "))
                        end
                    end
                end
            end
            write(f, "\n")
        end
    end
end

function printInstance(n_jobs, n_processors, UB, duration, processor)
    println()
    println("Running instance with parameters:")
    println("Jobs: ", n_jobs)
    println("Processors: " , n_processors)
    println("duration: ", duration)
    println("processor ", processor)
    println()
end

function printResults(s, occupiedRanges)
    for i in eachindex(s)
        println("Processor ", i, ": ", s[i])
    end
    println()
    println("Occupied ranges: ")
    for i in eachindex(occupiedRanges)
        println("Job ", i, ": ", occupiedRanges[i])
    end
    println("Objective: ", cost(s))
end
```