SNic SOLUTIONS
Operational Excellence Delivered

Manpower Coding Challenge

This is a coding challenge designed to assess software development and problem-solving skills.

You will be assessed on how well the solution meets the business requirements, the quality and maintainability of the code and overall stability and performance of the solution.

The reviewer should be able to check out your code and easily build and run the solution. Please provide notes if there are any additional steps required to do that.

Objectives

You will be given access to a private Github repository containing a partially-complete project. Your objective is to fork the repository, complete the project, and create a pull request back to the original repository with your solution.

The existing Program.cs file contains comments marked TODO that identify the main tasks you are expected to complete.

1. The first objective is to develop a manpower planning class that assigns work to employees based on their skills.

2. The second objective is to write the results out to a .CSV file. A sample of the expected format is provided in the repository in a file called assignments-sample.csv
   You should run your finished application against the two provided datasets – small and large. Each will create its own output file. These resulting files should be committed to the repository and submitted as part of your pull request.

In completing these tasks, you should also perform any other supporting activities that you feel are necessary. You are free to change the existing code and add new code as you see fit.

The existing code is not covered by unit tests. You do not need to add tests for existing elements of the application (but you may chose to add unit tests to cover the new logic that you are adding).

We encourage you to employ and demonstrate good development practices as these will be evaluated alongside the resulting solution.

3. Finally add a readme.md file to the repository containing notes on your approach to solving the problem, details of any algorithms or heuristics you employed, how you verified your results, etc.

Timeframe

It shouldn’t take more than a few hours to complete the project. There is no explicit limit on the time you can spend.

Once you have completed the pull request, we will review it and schedule a session to discuss the results.

Technical notes

The application you will extend is a .NET Core 2.2 console application.
It can be developed, built, and run on Windows, Linux or MacOS.

You can use any tools and add any open-source libraries to the project to assist with the development.

The rest of this document outlines the fictional scenario on which this challenge is based and describes the customer’s requirements.

If the requirements are not clear you may ask for clarification at any time.

Scenario

Acme Industries, Inc. is a metal fabrication company that makes a variety of components for the automotive industry.

You have been asked to help them solve a challenge they have with labor.

In one of their departments they have six employees who work on a variety of machining tasks.
The department head assigns tasks to the individuals based on the skills each employee has and the complexity of the tasks.
This is a time-consuming process and management believes the employees are not being utilized as efficiently as they would like.

Their IT department started to build some software that automates the task assignments based on some predetermined rules, but it isn't complete and doesn't currently build. You have been asked to step in and complete the project.

The Domain

Tasks

The department receives a list of tasks that detail the work to be done. Each task takes one person a full day to complete and must be performed by someone with a certain skill.

Some of the tasks are marked as having priority and should be completed as early as possible.

Here are a couple of example tasks:

Task Id	Skill Required	Priority?
101	Operator I	
102	Specialist Machinist	Yes
103	Operator II	

Skills

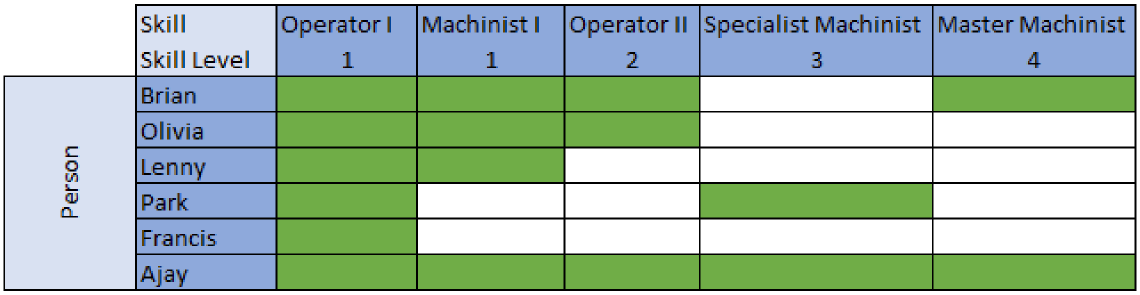
The following is a list of all the skills:

Skill	Skill level
Operator I	1
Machinist I	1
Operator II	2
Specialist Machinist	3
Master Machinist	4

The skill level is an indication of the relative difficulty of obtaining that skill and roughly correlates to the scarcity of the skill and, therefore, the cost of employing someone with that skill.

People

The department has six employees. Each employee is designated to have one or more skills as documented in the following skills matrix:



	Skill	Operator I	Machinist I	Operator II	Specialist Machinist	Master Machinist
	Skill Level	1	1	2	3	4
Person	Brian	🟩	🟩	🟩		🟩
	Olivia	🟩	🟩	🟩		
	Lenny	🟩	🟩			
	Park	🟩			🟩	
	Francis	🟩				
	Ajay	🟩	🟩	🟩	🟩	🟩

Employees can only work on tasks that require a skill that they possess.

As an example, since Francis only has the Operator I skill, she can only work on tasks that require Operator I.
Ajay, on the other hand, has all five skills and so could work on any task.

The Manpower Plan

The plan you are being asked to generate needs to identify which tasks will be performed on which day and by whom.

An example may look like this:

Day	Task	Person
1	101	Brian
	103	Olivia
	107	Lenny
	105	Park
	106	Francis
	108	Ajay
2	102	Brian
	104	Olivia
	109	Lenny
etc.		

Some additional information that will help generate a valid plan:

- Each task takes a full day to perform.
- Each task requires a single person to perform it.

These are metrics that management are interested in when evaluating the plan:

1. The number of days it takes to complete all the high priority jobs
2. The number of days it takes to complete the whole list of jobs
3. How effectively each person's skill set it being utilized
4. How long it takes to generate the plan

Current State of the Application

The application is partially written and already has the following functionality:

1. The user is prompted to choose between one of two datasets – a small data set containing a handful of tasks and a larger data set with around 300 tasks.

2. On execution it loads the tasks, people and skills from a series of .CSV files into an in-memory model.

   The path to each data file is read from a settings file called config.json
   The classes that make up the model exist in the Domain folder

3. The manpower planning process is invoked to generate the sequence of task assignments.
   This part of the application doesn't currently compile since the planning class is missing – you will have to create this class. See the comments in Program.cs

4. As the application runs it outputs information to the console to show progress.

5. Once the results of the manpower planning process have been calculated they are written out to a .CSV file on disk to be consumed by another system.
   This part of the application doesn't currently compile since the save logic is missing – you will have to extend the repository class to add this. See the comments in Program.cs