Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT KRPSIM (/PROJECTS/42CURSUS-KRPSIM)

You should evaluate 2 students in this team

★

Git repository

```
git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-80be911
```

Guidelines

This is the KrpSim grading system. Keep in mind that only a few display require an accurate formatting. The main focus of this project is ressource optimisation, hence, the efficiency of your program. On a similar process description file, the program may not systematically give the same answer each time you run it. This should not be a decisive factor.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/13362/en.subject.pdf)

resources.tgz (/uploads/document/document/2117/resources.tgz)

Preliminaries

Basics

Let's look at the basic obligations:

- the entire group is here and on time.

- the repo is complete.
- it includes 3 elements: the krpsim program, the krpsim_verif program, at least one of your own process files (several is better).

If one of the conditions has not been met, evaluation stops.

☑ Yes ☒ No

The swiss army lib

Make sure no module/library has been set to make the optimization job for you. You will be graded in regards of the time you've spent thinking about the subject, find answers by yourself and impplement your own solutions. This is not an integration project. If an external element is responsible for solving the problem, evaluation stops. The group goes back to work and tries harder this time.

☑ Yes ☒ No

Simple tests

Simple

We use simple tests, ideally provided with resources. But other kinds of tests are also fine. It's a simple chain, with a starting stock that is consumed and triggers processes one after the other. Does the krpsim reach the end of the process? Everytime? When you run the tests, keep in mind a short timeframe will give less opportunities to optimize or reach the objective. In contrast, a longer timeframe will make the objective more reachable (whether it's an accurate result or a better optimization) more often.
You can run a test with the krpsim_verif, but this should not be necessary for these simple tests. Your own process description files must, depending on their natures and difficulties, be used for the sections/questions that suit them best.

☑ Yes ☒ No

Simple returns

Using the same process file, do you always reach the end in the minimum time? Everytime? Minimum time must logically be the simple addition of

each process duration (since it's a simple process chain).

☑ Yes ☒ No

Ikea

Now, let's try an Ikea type configuration. Once again, it's better if there's a little more than an Ikea file. The goal, here, is to repeat and/or parallelize some actions to reach the objective. The stock should be used wisely to get all the necessary elements. There are only two stairs of process (all the stocks' possible paths from beginning to end include no more than 2 processes). Do we get the result each time (even if it's not completely accurate, once again)?

☑ Yes ☒ No

Ikea also returns

Is the time taken to reach the goal of Ikea (or any other similar file) otipimized? => ikea is achieved in 50 cycles for instance.

☑ Yes ☒ No

More complicated

This time, test a simple configuration containing at least one dead branch that consumes one resource (and cripples or stops the goal's achievement). Does it work? Do we reach the goal each time or so?

☑ Yes ☒ No

Fancy steak?

Test the steak file. If your krpsim is right, it should cook the whole thing in 30 sec every time.

☑ Yes ☒ No

Checking errors

syntax error

A file with misformed process/stock/optim: an error is displayed every time.

☑ Yes ☒ No

Missing parts

Are there any missing parts in te process description file ?

☑ Yes ☒ No

krpsim_verif

Make sure the krpsim_verif program displays an error message when the tranmitted file is not right. Also make sure the krpsim_verif displays an error message when the transmitted file contains an action made impossible because of the stock condition in the cycle in question. Finally, does the krpsim_verif indicate that everything is OK if it is so with a data tracing?

☑ Yes ☒ No

Open processes

non-stop

Test the "recre" configuration or any other simple process file that's self-sustained. There must be potential blocking situations, occurences where no process can be executed, that your krpsim should know how to avoid (as always, the more time it has, the better he will optimize the goal in order to avoid those blockings).

☑ Yes ☒ No

Performance

Is the performance better if you're playing with the timeframe offered to the program? (if the group can explain the performance is already

optimized and they cannot do better, it's fine).

☑ Yes ☒ No

Bigger

Test with "apple" or any other process complex description, containing dead branches, loops (like the eggs), potential blocking situations... well, a lot of difficulties, that is :) Check with the assessor. Check with the controller.

☑ Yes ☒ No

Bonus

Performance

If you have already made this project, use your own krpsim to compare performances, for instance. Otherwise, here is a possible performance scale for apple after 50000 cycles:

- less thant 50 000 euros (no box sold) : 0pt
- up to 100 000 euros
- up to 200 000 euros
- up to 500 000 euros
- up to 1M euros
- higher than 1M euros : 5pt. This scale will probably change along with the evaluation and the various projects' results.

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don’t forget to check the flag corresponding to the defense

✔ Ok ★ Outstanding project

Empty work No author file Invalid compilation Norme Cheat Crash

Incomplete group ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation