

Nome: _____

Instruções:

1. Pode ser feita consulta porém não é permitido copiar códigos fonte **que contenham uma parte substancial da resposta do exercício** (exceção: códigos fornecidos pelo professor);
2. Exercício INDIVIDUAL: Não pode consultar o colega. O resultado deve ser submetido no SIGAA.
3. Valor da avaliação: 1,5

Questão 1. (0,5) Implemente o algoritmo de multiplicação de matrizes utilizando o modelo mestre-escravo como visto em sala. Considere que as matrizes são quadradas e de tamanho 3. Faça um programa **mestre** que recebe como parâmetro (via argc, argv) um número de porta inicial **P**. Este mestre envia aos 3 escravos, conectados em portas **P**, **P+1** e **P+2**, utilizando Sockets e o protocolo UDP, os seguintes dados.

- A linha da Matriz A correspondente ao escravo (linha 0 para o primeiro escravo, linha 1 para o segundo e linha 2 para o terceiro)
 - Toda a Matriz B
- ✧ Dica: envie uma linha da Matriz B por vez!

Implemente também um código C que representa um **escravo** que recebe como parâmetro (também utilizando argc, argv) o número de porta que ele utilizará na transmissão. Utilizando sockets e protocolo UDP, o escravo recebe as informações enviadas pelo mestre (ver acima) e calcula uma linha da matriz C resultante e a envia de volta para o mestre. O mestre, por sua vez, imprime a matriz C final.

Lembre-se que para executar esta multiplicação de matrizes é preciso executar **UM** mestre e **3** escravos.

Questão 2. (0,5) Faça uma solução similar ao da questão anterior, porém agora o tamanho da matriz (e por consequência o número de escravos) serão passados por parâmetro na execução do mestre.

Faça um programa **mestre** que recebe como parâmetro (via argc, argv): (1) Um número de porta inicial **P** e (2) um valor inteiro **N**. Considere agora que as matrizes são quadradas e de tamanho **N**. Este mestre envia aos **N** escravos, conectados em portas **P**, **P+1**, **P+2...P+(N-1)**, utilizando Sockets e o protocolo UDP, os seguintes dados:

- O valor de **N**.
 - A linha da Matriz A correspondente ao escravo (linha 0 para o primeiro escravo, linha 1 para o segundo, etc)
 - Toda a Matriz B
- ✧ Dica: envie uma linha da Matriz B por vez!

Implemente também um código C que representa um escravo que recebe como parâmetro (também utilizando argc, argv) o número de porta **P** que ele utilizará na transmissão. Utilizando sockets e protocolo UDP, o escravo recebe as informações enviadas pelo mestre (ver acima) e calcula uma linha da matriz C resultante e a envia de volta para o mestre. O mestre, por sua vez, imprime a matriz C final.

Lembre-se que para executar esta multiplicação de matrizes é preciso executar **UM** mestre e **N** escravos. A execução dos escravos deve ser feita em separado em terminais diferentes. Logo, se o planejado é executar o mestre passando o valor **N** igual a 5, devem ser executados 5 escravos.

Questão 3. (0,5) Faça uma solução similar ao da questão anterior (questão 2), porém o programa escravo recebe como parâmetro (via argc, argv): (1) Um número de porta inicial **P** e (2) um valor inteiro **N**. **N** representa (além do tamanho das matrizes, como na questão anterior) o número de **THREADS** a serem criadas e agora cada thread é um escravo que irá calcular a linha da matriz C correspondente (assim como explicado nas questões anteriores). O código escravo é executado apenas uma vez (passados os devidos parâmetros) e são criadas **N** threads que conectam ao mestre nas portas **P**, **P+1**, **P+2...P+(N-1)**, utilizando Sockets e o protocolo UDP.

Lembre-se que para executar esta multiplicação de matrizes é preciso executar **UM** mestre e **UM** escravo. A diferença é que agora o escravo executado cria **N** threads que irão atuar exatamente como os escravos da questão anterior.