Esta prova tem quatro folhas; leia e siga as instruções I, II e III antes de iniciar a mesma.

I Como fazer a prova

Há quatro questões nesta prova; todas elas são especificações de programa que você deve implementar, isto é: escrever o código-fonte, compilar e executar. Teste bem o seu programa; esteja atento às restrições de cada questão. Cada questão, deve possuir dois arquivos como resposta: **.c** e **.pdf**.

O arquivo **.c** deve ter APENAS o código-fonte correspondente à sua resposta. O topo desse arquivo deve ter seu nome e número de matrícula.

O arquivo **.pdf** tem de ser gerado por um editor de texto que lhe permita inserção de uma imagem. Essencialmente, o conteúdo desse arquivo é o mesmo daquele do arquivo .c com a fonte **Consolas** ou **Lucinda Console** ou **Courier**; porém no topo do arquivo, você deve inserir

- seu nome e seu número de matrícula;
- a Declaração Idoneidade da Prova, abaixo;
- E uma imagem de sua assinatura.

O nome dos arquivos depende da questão que está sendo respondida. Se o conteúdo do arquivo corresponder à **resposta da questão 1**, então o nome dos arquivos serão **q1.c** e **q1.pdf**; se for resposta à questão 2, então o nome dos arquivos serão q2.c e q2.pdf; e assim por diante.

Declaração de Idoneidade da Prova

"Eu afirmo que todas as respostas desta prova são de minha autoria. Durante todo o tempo em que esta prova esteve comigo, isto é, até antes de eu devolver esta prova no Teams©, eu não conversei por meio escrito ou oral, por qualquer meio de comunicação, com qualquer outra pessoa (física ou jurídica) sobre qualquer questão desta prova."

Cole aqui sua assinatura

II Atentar para o prazo de entrega da prova no Teams

Esta prova se tornou disponível às 14h de 2 de outubro de 2020; o **ENCERRAMENTO** da mesma se dará às **23h59 de 3 de outubro de 2020**. O Teams© será configurado para NÃO ACEITAR entregas após esse horário. Contudo, se de alguma forma entrega da prova com atraso ocorrer, a nota da mesma terá decréscimo de 80%.

Quando você decidir que concluiu a prova **OU** o tempo de prova estiver em seus 5 minutos finais, suba os arquivos .c e .pdf para a sala virtual GBC014 2020/1Esp, na Seção Tarefas, de modo similar ao que você vem fazendo quando entrega uma Pós-Aula.

III As repostas das questões

Em cada arquivo, basta colocar a resposta de cada questão; não há necessidade de reproduzir o texto da questão.

Os arquivos .c entregues serão submetidas ao compilador GCC de um sistema operacional GNU/Linux; especificamente, irei corrigir a prova no WSL que tem Ubuntu instalado. Por isso, a necessidade do envio dos arquivos .c, contudo, a questão não será corrigida e ganhará zero ponto, se o arquivo .pdf correspondente não for juntamente enviado.

As respostas receberão o total de pontos da questão, se somente se:

- o código-fonte do programa não tiver erro de compilação;
- E o código-fonte do programa não tiver *warnings* de compilação;
- E a execução do código executável, gerado pelo compilador, se comportar tal como especificado na questão;
- E o código-fonte do programa respeitar as restrições da questão;
- E o código-fonte do programa estiver em conformidade com o Padrão de Codificação da disciplina, que está divulgado no Teams©.

QUESTÕES DE 1 A 4

1. (5 pontos) Escreva um programa que apresenta em stdout (tela do computador) o conteúdo de um arquivo. O nome do arquivo é informado pelo usuário do programa. O programa termina quando todo o conteúdo do arquivo tiver sido apresentado.

2. (5 pontos) Escreva um programa que grava em um arquivo tudo o que o usuário escreve em `stdin` (digita no teclado). O nome do arquivo é informado pelo usuário do programa. O programa termina quando o usuário digita `<enter> (\n)`.

3. (10 pontos) Escreva um programa que apresenta em `stdout` sequências de caracteres concatenadas pelo caractere asterisco (*).

Restrições

- As sequências estão contidas em um arquivo; o nome do arquivo é apresentado pelo usuário do programa.
- O arquivo contém várias linhas; cada linha contém sequências de caracteres separadas por espaço. A quantidade de linhas do arquivo não é conhecida pelo programador. Um exemplo hipotético de arquivo é apresentado abaixo.

```
Fulano das Couves 2000
Vote 2010: Beltrano da Chicória
Sicranos Tomateiro! Nesse você confia!!!
```

- As sequências contidas no arquivo podem ser constituídas por qualquer caractere do teclado: letra, algarismo, símbolos (&, #, etc). Para um arquivo tal como acima, o programa deveria apresentar o que segue abaixo.

```
Fulano*das*Couves*2000
Vote*2010:*Beltrano*da*Chicória
Sicranos*Tomateiro!*Nesse*você*confia!!!
```

- Não é permitido o uso de *array* para armazenar as linhas do arquivo.
- O programa termina quando todas as linhas do arquivo tiverem sido apresentadas.

4. (15 pontos) Escreva um programa que apresenta o conteúdo de uma matriz N×N e dados sobre essa matriz.

Restrições

- A matriz é de números inteiros e deve ser implementada por um array bidimensional 10×10.
- Os valores da matriz estão armazenados em um arquivo em disco.
  - O nome desse arquivo é informado pelo usuário do programa.
  - O arquivo é constituído por algarismos e espaço; uma sequência de algarismos representa um número; cada sequência é separada por um espaço.
  - A primeira sequência representa a ordem da matriz; as demais sequências representam os números da matriz.
  - Um exemplo hipotético de conteúdo de arquivo é apresentado abaixo

    ```
    3 10 10 10 20 20 20 30 30 30
    ```

    3 representa a ordem da matriz; as 9 sequências restantes representam os números que deverão preencher a matriz.
  - O conteúdo do arquivo é bem comportado, portanto, SEMPRE: a ordem N da matriz será 2 ≤ N ≤ 10; a quantidade de sequências corresponderá exatamente quantidade de posições da matriz.
- Seja qual for o valor de N, o preenchimento do *array*, com os dados do arquivo, deve seguir a ordem das linhas: primeiro preenche a primeira linha, depois a segunda linha e assim por diante.
- O programa deve apresentar a matriz em ordem de linha: primeira linha, depois a segunda linha e assim por diante. Considerando uma matriz de ordem 3, tal como o conteúdo de arquivo apresentado no exemplo (vide acima), o programa deverá apresentar essa matriz como abaixo.

  ```
  10   10   10
  20   20   20
  30   30   30
  ```
- Após a apresentação da matriz, o programa torna a interagir com o usuário. O programa apresenta opções para o usuário: DP, DS, TotalLinha, Sair.
  - A opção **DP** faz o programa apresentar a Diagonal Principal da matriz. Essa apresentação deve ser realizada pela função `dp()`. Essa função recebe o tamanho do *array* e o *array* que contém a matriz, e retorna nada.
  - A opção **DS** faz o programa apresentar a Diagonal Secundária da matriz. Essa apresentação deve ser realizada pela função `ds()`. Essa função recebe o tamanho do *array* e o *array* que contém a matriz, e retorna nada.
  - A opção **TotalLinha** apresenta a soma dos valores contidos em uma linha da matriz. Essa apresentação deve ser realizada pela função `tl()`. Essa função recebe o tamanho do *array*, o *array* que contém a matriz, e o um número X correspondente à linha de interesse; e retorna o valor correspondente à soma dos valores na linha X.
    - O valor de X é informado pelo usuário do programa. Quando o usuário selecionar a opção TotalLinha, o programa solicita em seguida que o usuário digite o número da linha desejada.
    - Em tempo: o usuário é bem comportado (ele não irá digitar um número de linha que inexiste na matriz); o usuário entende de matriz, não entende de *array*; então quando o usuário digitar 1, o programa deve considerar a linha 0 do *array*.

- A opção **Sair** termina o programa. Portanto, o programa termina APENAS quando o usuário selecionar esta opção.