

1) Considere a matriz $A = [a_{ij}]_{n \times m}$, onde $n = 4$ e $m = 5$, com número inteiros gerados aleatoriamente de 1 até 20. Faça um algoritmo para gerar a matriz A e verificar se ela satisfaz a seguinte condição:

$$\min_{1 \leq j \leq m} \sum_{i=1}^n |a_{ij}| \leq \max_{1 \leq i \leq n} \prod_{j=1}^m a_{ij}$$

Crie e utilize uma **função** para gerar a matriz e outra para realizar a verificação. De acordo com o retorno da função de verificação, deve-se imprimir na função *main*: “Condicao Satisfeita” ou “Condicao Nao Satisfeita”.

2) Considere uma matriz M de ordem 4 de números inteiros gerados aleatoriamente de 0 até 29. Faça um algoritmo para gerar esta matriz e imprimir na tela se ela é ou não uma **Matriz Ortogonal**.

Crie e utilize quatro **funções**: uma para gerar a matriz M , outra para calcular a sua Matriz Transposta (M^T), outra calcular a multiplicação $M \times M^T$ e a quarta para retornar se a matriz M é Ortogonal ou não. A impressão desta informação tem que ser na função *main*.

Obs.: Se uma matriz quadrada M é uma matriz ortogonal, então $M \times M^T = I$, onde M^T é a Matriz Transposta de M e I a Matriz Identidade.

3) Considere um vetor que armazena 10 números inteiros pares e 10 números inteiros ímpares todos embaralhados, ou seja, sem qualquer ordem preestabelecida. Faça um algoritmo para ler este vetor do teclado e depois organizá-lo de modo que os números **pares** fiquem nas posições **ímpares** do vetor e os números **ímpares** fiquem nas posições **pares** do vetor.

Crie e utilize duas **funções**: uma para preencher o vetor pelo teclado e o outra para organizá-lo.

Obs.: Não é permitido utilizar qualquer estrutura de dados para auxiliar a organização.

4) Considere um vetor com 30 números inteiros gerados aleatoriamente de 1 até 30. Faça um algoritmo para gerar este vetor e depois ordená-lo de maneira **não-crescente**.

Crie e utilize três **funções**: uma para preencher o vetor, outra para ordená-lo e uma terceira para imprimir o vetor antes e depois da ordenação.

Obs.: Não é permitido utilizar qualquer estrutura de dados para auxiliar a ordenação.

5) Considere dois números inteiros a ($a \neq 0$) e b ($b \geq 0$) lidos pelo teclado. Faça um algoritmo **recursivo** para calcular o valor de a^b .

Crie e utilize uma **função recursiva** para calcular a^b . Esta informação tem que ser impressa na função *main*.

Obs.: Não é permitido utilizar qualquer estrutura de repetição na função recursiva.

6) Considere um vetor com 20 números naturais maiores do que 1 lidos pelo teclado. Faça um algoritmo **recursivo** que organize este vetor de modo que os números **compostos** fiquem nas **primeiras** posições e os números que **não são compostos** nas **últimas** posições.

Crie e utilize duas **funções**: uma para preencher o vetor e outra recursivo para realizar a organização do mesmo. Crie e utilize também outra **função** para retornar 1, se um número natural for composto, ou retornar 0, caso contrário.

Obs. 1: Um número natural C é composto se ele tem mais de dois divisores naturais distintos;

Obs. 2: Não é permitido utilizar qualquer estrutura de dados para auxiliar a organização;

Obs. 3: Não é permitido utilizar qualquer estrutura de repetição na função recursiva.

7) Considere um vetor com 50 números inteiros gerados aleatoriamente de 1 até 100. Faça um algoritmo **recursivo** para imprimir o **maior valor** deste vetor.

Crie e utilize uma **função** para preencher o vetor e uma **função recursiva** para encontrar o maior valor do vetor. Esta informação tem que ser impressa na função *main*.

Obs. 1: Não é permitido utilizar qualquer estrutura de dados auxiliar;

Obs. 2: Não é permitido utilizar qualquer estrutura de repetição na função recursiva.

Importante

- O grupo deve enviar os códigos (em uma pasta compactada) até às **23h59** do dia **14/12/2022** para o seguinte e-mail:

philippeal@yahoo.com.br

- No dia da apresentação (**15/12/2022**) o grupo deverá apresentar o trabalho para o professor, onde o mesmo fará perguntas sobre os algoritmos.
- Apenas um e-mail por grupo deve ser enviado. Isto é, não há a necessidade de cada componente do grupo enviar. No entanto, recomenda-se enviar uma cópia do e-mail para os demais componentes.
- O **ASSUNTO** do e-mail e o **NOME DO ARQUIVO** devem ter a seguinte formatação:

BSI-PP-ListadeExercicios-Linguagem-PrimeiroNomedosAlunos

Exemplo de e-mail do grupo (fictício) formado pelos alunos Lucas Pereira, Renato da Silva e Miguel dos Santos que farão o trabalho na Linguagem C:

Para: philippeal@yahoo.com.br

De: Lucas Pereira

Assunto: BSI-PP-ListadeExercicios-C-Lucas-Renato-Miguel

Anexo: BSI-PP-ListadeExercicios-C-Lucas-Renato-Miguel.c

Repare que o nome do arquivo (.c) deve ter a mesma formatação do “Assunto” do e-mail:

BSI-PP-ListadeExercicios-C-Lucas-Renato-Miguel.c

- E-mail com o “Assunto” fora do padrão **NÃO SERÁ ACEITO**.
- A primeira linha de cada código deve conter: os nomes dos **componentes do grupo**.
- Utilize nomes sugestivos para as variáveis. Faça corretamente a indentação e comentários no código para facilitar seu entendimento. Estes itens serão avaliados.
- Preste atenção se está enviando a versão correta do trabalho, visto que será considerada a data da entrega da versão correta.

- Não deixe para enviar o trabalho na última hora, pois podem acontecer problemas com o envio.
- Caso o grupo esteja com dúvidas, tire-as com o professor na aula **ANTES** da entrega e apresentação. Não deixe para a última hora.
- Os trabalhos serão avaliados utilizando o Sistema Operacional **Linux Mint 19.3**.

- Trabalhos entregues com atraso receberão uma penalidade P :

$$P = 0,9 - 0,2(d - 1)$$

$$\text{Nota Máxima} = \text{Nota do Trabalho} \times P$$

onde d ($d \geq 1$) é o número de dias em atraso. Exemplo: se o grupo entregar com 1 dia de atraso, ele poderá ficar com, no máximo, 90% da nota do trabalho. Repare que após 5 dias de atraso não é mais possível entregar o trabalho. O atraso é contado em número de dias e não de horas. Caso o grupo envie o trabalho a qualquer hora do dia seguinte a data de entrega, será considerado 1 dia de atraso, e assim por diante.