



→ **INSTRUÇÕES. LEIA ATENTAMENTE ANTES DE INICIAR A PROVA:**

1. A prova deve ser feita utilizando a linguagem C.
2. Todas as questões devem estar de acordo com as boas práticas de programação (comentários, indentação, etc).
3. A prova deve ser feita de forma **INDIVIDUAL**. Caso sejam identificadas cópias, todas serão sumariamente zeradas.
4. Crie uma pasta com o seu nome_sobrenome (Ex.: Manoel_Junior) e grave todos os arquivos dentro dela.
5. **NÃO ENVIE EXECUTÁVEIS.**
6. Salve cada questão com o seguinte formato `questaoX.c`, onde X é o número da questão. Ex.: `questao1.c`
7. Compacte a pasta criada com suas resoluções e envie para o local correto no Google Classroom.
8. O tempo de prova inclui a criação e gravação dos arquivos e pastas.
9. Caso seja necessário, escreva suas considerações em um arquivo texto, chamado de **CONSIDERACOES.txt** e envie junto das questões.

Questão 1 (50%)

Números triangulares (*tri*), pentagonais (*pen*) e heptagonais (*hep*) são dados, respectivamente, pelas fórmulas:

$$tri = \frac{n(n+1)}{2} \quad pen = \frac{n(3n-1)}{2} \quad hep = \frac{n(5n-3)}{2}$$

Dessa forma, os primeiros números de cada sequência são, respectivamente:

$$\begin{aligned} tri &= 1, 3, 6, 10, 15, \dots \\ pen &= 1, 5, 12, 22, 35, \dots \\ hep &= 1, 7, 18, 34, 55, \dots \end{aligned}$$

Faça um algoritmo, **obrigatoriamente usando sub-rotinas**, que verifique se existe algum número que faça parte dos três conjuntos.

Questão 2 (50%)

Faça um algoritmo, **obrigatoriamente usando sub-rotinas**, que preencha uma matriz 20x20 com valores pseudoaleatórios* entre 1 e 500. Em seguida, partindo da posição 0,0 (coluna zero e linha zero), seu algoritmo deve apresentar um caminho na tela conforme as regras abaixo:

- Caso o valor da célula atual seja divisível pelo menor número da respectiva linha, deve-se caminhar para a célula imediatamente à direita.
- Caso o valor da célula atual **não** seja divisível pelo menor número da respectiva linha, deve-se caminhar para a célula imediatamente abaixo.
- Se o algoritmo atingir a última linha, mas não a última coluna, o algoritmo deverá caminhar sempre para a direita, independente das regras anteriores, finalizando assim na última célula (19,19);

Observação: como resultado, seu algoritmo deve mostrar na tela o caminho percorrido, usando D para direita e B para baixo, além de mostrar o resultado da soma dos elementos do caminho feito. Veja o **EXEMPLO** abaixo (para uma matriz de tamanho reduzido). Note que o 377 não é divisível por 37, mas como está na última linha obrigatoriamente foi necessário caminhar para a direita.

498	176	386	83	228
10	70	204	33	282
309	61	174	152	17
450	3	261	398	165
489	37	428	377	193

Saída correta do algoritmo para a matriz exemplo acima: Caminho = DBDBBDBD | Soma = 2351

* veja como trabalhar com números pseudoaleatórios aqui: <http://linguagemc.com.br/valores-aleatorios-em-c-com-a-funcao-rand/>