



Pontifícia Universidade Católica de Minas Gerais  
 Bacharelado em Engenharia de Software - Unidade Praça da Liberdade  
 Programação Modular e Laboratório de Programação Modular  
 Prof. João Caram - Semestre 1/2024  
 Exercício de Revisão (autodiagnóstico) e Nivelamento (Desafio Java)

Esta é a primeira atividade das disciplinas Programação Modular (PM) e Laboratório de Programação Modular (LPM). Ela serve como autodiagnóstico na disciplina teórica (PM) e será pontuada de duas formas na disciplina prática (LPM). Seus objetivos principais são, primeiro, fazer uma revisão de Algoritmos e técnicas de programação básicas e, segundo, promover a transição para a linguagem Java sem necessariamente utilizar os conceitos de orientação por objetos neste momento. Atenção para as instruções:

1) Utilize o mecanismo web de busca de sua preferência para procurar por "300 ideias para programar". Localize, assim, o PDF do material "**300 Ideias Para Programar Computadores**", de Virgílio Vasconcelos Vilela.

### **Básicos**

**1.1.6 Quadrado** - Escrever um programa que mostre a seguinte figura no alto da tela:

```
XXXXXX
X      X
X      X
X      X
XXXXXX
```

**1.1.10 Palavra grande** - Escreva um programa que produza a tela esquematizada abaixo:

```
*****  *****  *
*          *      *  *
*          *      *  *
*****  *      *  *
          *      *  *
          *      *  *
*****  *****  *****
```

**1.1.15 Menu** - Elabore um programa que mostre o seguinte menu na tela:

```

Cadastro de Clientes

0 - Fim
1 - Inclui
2 - Altera
3 - Exclui
4 - Consulta

Opção: _

```

**1.2.5 Otimização de corte** - Um marceneiro, para fazer um trabalho, precisa cortar vários pedaços de madeira de 45 cm cada um. Ele pode comprar tábuas de 3, 4 ou 5 metros. Usando os operadores **div** e **mod**, faça um programa que calcule a quantidade de pedaços e a sobra para cada tipo de tábua, permitindo assim uma melhor escolha do marceneiro.

**1.2.8 Imposto** - Um imposto é calculado com base na seguinte tabela:

Até	1.200,00	isento
de	1.201,00 a 5.000,00	10%
de	5.001,00 a 10.000,00	15%
acima de	10.000,00	20%.

Implemente um programa que calcule os impostos a pagar para um valor em cada faixa. Para cada um, mostre uma mensagem que identifique na tela a que se refere cada valor.

**1.2.12 Cálculo de notas** - Um professor atribui pesos de 1 a 4 para as notas de quatro avaliações. A nota é calculada por meio da média ponderada  $(N1 + N2*2 + N3*3 + N4*4)/10$ , onde N1 é a nota da primeira avaliação, N2 a da segunda, etc. Um aluno tirou as seguintes notas: 8 - 7,5 - 10 - 9. Faça um programa que calcula e mostra as notas e a média deste aluno, sendo a média formatada com 1 casa decimal.

**1.2.13 Funções aninhadas** - Escrever um programa que calcula a raiz de 3, arredonda o resultado e calcula a exponencial do valor resultante.

**1.2.14 Devagar se vai ao longe** - Vou e volto diariamente a pé para o trabalho, que dista aproximadamente 800 m de minha casa. Supondo que trabalho 5 dias por semana, 45 semanas por ano, "bole" a operação matemática que deve ser efetuada para calcular quantos quilômetros, aproximadamente, terei andado ao final de um ano. Elabore um programa que faça as contas e mostre o resultado na tela.

## 2.Variáveis e Entrada de Dados

**2.1.3 Losangos 1** - Implemente um programa que desenhe os losangos abaixo na tela, sendo que o topo do primeiro losango é colocado em uma linha e uma coluna lidas do teclado, e o topo do segundo fica 15 colunas à direita do primeiro.

```

      X
     XXX
    XXXXX
   XXXXXXX
  XXXXXXXXX
 XXXXXXXXXXX
XXXXXXX
 XXXXX
  XXX
   X

```

```

      X
     XXX
    XXXXX
   XXXXXXX
  XXXXXXXXX
 XXXXXXXXXXX
XXXXXXX
 XXXXX
  XXX
   X

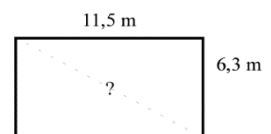
```

**2.2.8 Tempo livre** - Reescreva o programa 1.2.4 (o estudante metódico) de forma que trate qualquer disciplina e qualquer quantidade de tempo livre. Assim, o estudante entra com esses valores e o programa efetua os cálculos necessários.

**1.2.4 Tempo livre** - Um estudante muito metódico estava matriculado em 6 disciplinas, e dispunha de 1 hora e 40 minutos para estudar. Sua intenção era dividir o tempo disponível igualmente para as 6 disciplinas, e descansar livremente o tempo restante. Faça um programa que calcule o tempo que ele deve dedicar para cada disciplina e o tempo livre. [Dica: use os operadores **div** e **mod**]

**2.2.9 Comprimento de fio** - Altere o programa do eletricitista (1.2.10) para que as medidas sejam lidas do teclado.

**1.2.10 Comprimento de fio** - Um eletricitista precisa comprar fio que irá passar, pelo telhado, por toda a diagonal de uma casa de formato retangular. Como ele não tem condições de medir a diagonal com precisão (ou talvez não queira...), a solução alternativa que ele encontrou foi medir os lados da casa, sabendo que a diagonal pode ser calculada com base nos lados pelo Teorema de Pitágoras ( $a^2 = b^2 + c^2$ ). Considerando que a casa mede 11,5 x 6,3 metros, faça um programa que calcule a quantidade mínima necessária de fio a ser comprada, com precisão até centímetros.



**2.2.12 Despesas de casal 1** - Um casal divide as despesas domésticas mensalmente. Durante o mês cada um anota seus gastos e as contas que paga; no final eles dividem meio a meio. O casal deseja um programa que facilite o acerto: eles digitariam os gastos de cada um, e o programa mostraria quem deve a quem. Atualmente eles fazem o acerto manualmente, na forma da seguinte tabela:

ITEM	MARIDO	ESPOSA	TOTAL
DESPESAS PAGAS	1278,60	875,30	2.153,90
% PAGO	59,36	40,64	100
VALOR DEVIDO	1.076,95	1.076,95	2.153,90
SALDO	201,65	-201,65	

Portanto, os saldos devem ser iguais, e quem tiver o saldo negativo deve pagar o valor para o outro. Faça um programa que leia os valores adequados e efetue os cálculos. O total é a soma das despesas individuais; um percentual é o gasto individual dividido pelo total, multiplicado por 100; o valor devido por cada um é o mesmo e igual à metade do total; finalmente, cada saldo corresponde à metade da diferença entre o valor pago pela pessoa e o valor total. Uma tela para o programa pode ser, com os mesmos dados da tabela acima:

Digite valor das despesas do marido: 1278.60			
Digite valor das despesas da esposa: 875.30			
ITEM	MARIDO	ESPOSA	TOTAL
=====	=====	=====	=====
Despesas pagas	1278.60	875.30	2153.90
% pago	59.36	40.64	100
Valor devido	1076.95	1076.95	2153.90
Saldo	201.65	-201.65	

**2.2.14 Despesas de casal 3** - Para o mesmo programa de rateio acima, suponha que o casal, ao invés de dividir meio a meio as despesas, vai dividi-las proporcionalmente à renda de cada um. Altere o programa de forma que este leia também a renda de cada um e use a proporção das rendas para a divisão.

**2.2.15 Adivinha** - Escrever um programa que “adivinha” o número pensado por uma pessoa (Pense um número (pausa), multiplique por 2 (pausa), some 6 ao resultado (pausa), divida o resultado por 2, quanto deu? (informe o resultado), você pensou o número tal). [Dica: problemas desse tipo dão origem a uma expressão aritmética, e você pode alterar as operações à vontade, desde que a expressão resultante admita uma inversa. Normalmente estruturamos o problema de forma que a expressão permita uma simplificação que facilite os cálculos. Para a sequência proposta, a expressão é (sendo n o número pensado e R o resultado):  $(n*2+6)/2 = R$ , donde  $n = (R*2-6)/2 = R - 3$ . Ou seja, basta subtrair 3 do resultado fornecido pela pessoa para "adivinhar" o número].

**2.2.19 Notas do professor** - Um professor avalia seus alunos através dos seguintes critérios:

- a) Duas notas de exercícios de 0 a 10, sem casas decimais, peso 1 e peso 2, respectivamente, com peso de 20% na nota final.
- b) Uma nota de prova de 0 a 10, com uma casa decimal e peso de 80% na nota final.

Elabore um programa que lê as notas de um aluno, calcula e mostra na tela sua nota final, formatada com uma casa decimal e devidamente ponderada pelos pesos (uma média ponderada é calculada somando-se os produtos de cada valor pelo seu peso e dividindo-se a soma resultante pela soma dos pesos). Exemplo: Um aluno tirou 5 e 6 de exercícios e 8,5 na prova. Sua nota de exercícios é  $(5*1 + 6*2) / 3 = 5,667$ . Sua nota final é  $(5,667*2 + 8,5*8) / 10 = 7,9$ .

**2.2.20 Conta telefônica** - Uma conta telefônica é composta dos seguintes custos:

- assinatura: R\$ 21,40
- impulsos: R\$ 0,03 por impulso que exceder a 90
- interurbanos chamadas p/ celular: R\$0,40 por impulso

Elabore um programa que lê os impulsos excedentes, valor de interurbanos e quantidade de chamadas para celular e calcula o valor da conta. Ao definir a tela, imagine que está fazendo um produto para ser avaliado por um cliente, juntamente com o de concorrentes, para uma eventual compra.

## 2.3. CARACTERES E CADEIAS

**2.3.4 Iniciais** - Escreva um programa que lê nome e sobrenome, e mostra na tela as iniciais.

**2.3.6 Metades de cadeia** - Implemente um programa que lê uma cadeia de caracteres de tamanho até 255 e mostra na tela as metades da cadeia. [*Dica: basear os cálculos no tamanho da cadeia*]

**2.3.9 Componentes de data** - Escrever um programa que lê uma data no formato 'dd/mm/aa' e mostra dia, mês e ano separados.

**2.3.10 Sorteio da LBV** - A LBV fez um sorteio cujos bilhetes continham números de 6 dígitos. O sorteio foi baseado nos dois primeiros prêmios da loteria federal, sendo o número sorteado formado pelos três últimos dígitos do primeiro e do segundo prêmio. Por exemplo, se o primeiro prêmio fosse 34.582 e o segundo 54.098, o número da LBV seria 582.098. Escreva um programa que lê os dois prêmios e retorna o número sorteado.

## Alternativas e Decisão

## 3.2. MATEMÁTICA

**3.2.2 Maior de 3** - Faça um programa que lê três números diferentes e mostra na tela uma mensagem indicando qual é o maior.

**3.2.4 Aprovação 1** - Elaborar programa que lê uma disciplina e respectiva nota final, múltipla de 0,5, e informa o que ocorreu. Se a nota for de 5 a 10, aprovado; se 4 ou 4,5, segunda época e, caso contrário, reprovado.

**3.2.6 Aprovação 3** - Altere o programa acima para que leia também a quantidade de aulas ministradas e a quantidade de faltas do aluno. Se o aluno não obteve 75% de frequências, ele está reprovado, independentemente da nota.

**3.2.8 Conta telefônica** - Uma conta telefônica é composta dos seguintes custos:

assinatura:	R\$ 17,90
valor de impulsos:	R\$ 0,04 por impulso que exceder a 90
valor de interurbanos	
valor de chamadas p/ celular:	R\$0,09 por impulso

Elabore um programa que lê valor de interurbanos, quantidade total de impulsos normais e para celular, e calcula o valor da conta.

**3.2.9 Tipo de triângulo** - Em um triângulo, cada lado é menor do que a soma dos outros dois. Escreva um programa que lê três valores e informa se estes não podem constituir um triângulo ou, caso contrário, se o triângulo formado é equilátero (três lados iguais), isósceles (dois lados iguais) ou escaleno (lados diferentes).

**3.2.10 Salário** - Um salário tem os seguintes componentes:

- Valor nominal
- Adicional devido a horas extras
- Valor descontado para o INSS (10% do valor a receber, limitado a 150 reais).

O valor adicional devido às horas extras é calculado dividindo-se o valor nominal por 176 (22 dias de 8 horas), multiplicando-se pela quantidade de horas e ainda com um acréscimo de 50%.

Escrever um programa que lê os valores necessários, calcula e mostra na tela os componentes do salário e o salário líquido resultante para o empregado. Não é preciso prever arredondamentos, mas os valores devem ser mostrados na tela com duas casas decimais.

Exemplos: para um salário de R\$ 1.000,00, com 30 horas extras, teremos R\$ 255,68 de horas extras  $[(1.000/176) * 30 * 1,5]$ , R\$ 125,57 de INSS e um salário líquido de R\$ 1.130,11. Para um salário de R\$ 2.000,00 e 20 horas extras, seriam R\$ 340,91 de horas extras, R\$ 150,00 de INSS (e não os 10%), com um salário líquido de R\$ 2.190,91.

**3.2.14 Imposto** - Um imposto é calculado com base na seguinte tabela:

Até 1.200,00	isento
de 1.201,00 a 2.500,00	10%
de 2.501,00 a 5.000,00	15%
acima de 5.000,00	20%.

Implemente um programa que leia o valor base e calcule o imposto a pagar.

## CARACTERES E CADEIAS

**3.3.4 Validação de data** - Escrever um programa que lê uma data no formato 'DD/MM/AAAA' e verifica se as barras estão nas posições corretas, se o dia está entre 1 e 31 e se o mês está entre 1 e 12, mostrando mensagens de erro apropriadas ou que a data está correta.

**3.3.6 Tipo de caractere** - Escrever um programa que lê um caractere e informa se é letra, dígito, operador aritmético ou nenhum deles.

## VARIADOS

**3.6.2 Dia da semana** - Construa um programa que lê um número de 1 a 7 e informa o dia da semana correspondente, sendo domingo o dia de número 1. Se o número não corresponder a um dia da semana, é mostrada uma mensagem de erro.

**3.6.3 PIS/PASEP** - O dígito verificador do PIS/PASEP é calculado através da seguinte regra: o número é composto por dez dígitos mais um dígito verificador. Multiplique os números, da esquerda para a direita, respectivamente por 3 2 9 8 7 6 5 4 3 2. Some os resultados das multiplicações; calcule o resto da divisão da soma por 11 e subtraia o resultado de 11. Se o resultado for 10 o dígito é zero, caso contrário o dígito é o próprio resultado.

Por exemplo, para o número 1701209041-1, o cálculo seria:

$1 \times 3 + 7 \times 2 + 0 \times 9 + 1 \times 8 + 2 \times 7 + 0 \times 6 + 9 \times 5 + 0 \times 4 + 4 \times 3 + 1 \times 2 = 98$ . O resto da divisão de 98 por 11 é 10. Como  $11 - 10 = 1$ , o dígito é 1.

Escreva um programa que lê um número de PIS/PASEP e mostra o dígito verificador correspondente. Para testá-lo, você pode usar também o número 1010861269-1.

**3.6.4 Calculadora** - A calculadora de Luciana pifou, justo quando ela precisa fazer vários cálculos. Ela tem um computador, mas não sabe que um dos acessórios do Windows é uma calculadora. Sendo estudante de programação, Luciana resolveu fazer um programa. A especificação que bolou prevê que programa lê dois números inteiros (o que atende suas necessidades) e em seguida um símbolo de operação. Se este for '+', o programa soma os números, se '-', subtrai, se '\*' multiplica e se '/' divide. Se o símbolo for diferente desses, é mostrada uma mensagem de erro. O programa, antes de dividir, critica se o divisor é zero e mostra uma mensagem, se for. Implemente a especificação de Luciana.

**3.6.5 Jogo de fichas 1** - Um jogo consiste em se retirar duas fichas de um saco contendo fichas brancas e pretas. Dependendo da combinação de cores das fichas retiradas, o jogador será pago na seguinte proporção:

<i>Primeira Ficha</i>	<i>Segunda Ficha</i>	<i>Rateio</i>
Branca	Branca	0
Branca	Preta	1/2
Preta	Branca	1
Preta	Preta	2

Ou seja, com duas fichas brancas o jogador perde tudo, com uma branca e uma preta recebe metade do que apostou, com um preta e uma branca recebe seu dinheiro de volta e com duas pretas recebe o dobro. Elaborar um programa que lê as cores das duas fichas e calcula o rateio.

**3.6.12 Multiplicação rápida** - Um algoritmo para multiplicação rápida por 11 de números de 2 dígitos funciona assim: para multiplicar  $81 \times 11$ , some os dígitos do número ( $8 + 1 = 9$ ) e insira o resultado entre os dígitos (891). Se a soma der maior que 9, incremente o dígito da esquerda (vai um):  $56 \times 11 = 616$ . Faça um programa que efetue multiplicações por 11 usando este algoritmo.

## Repetição

### MATEMÁTICA

**4.2.11 Ímpares múltiplos 1** - Escreva um programa que soma todos os números ímpares múltiplos de três situados na faixa de 1 a 1000.

**4.2.12 Ímpares múltiplos 2** - Altere o programa acima de forma que a faixa seja informada pelo usuário, e os números ímpares múltiplos de três sejam mostrados em ordem decrescente.

**4.2.15 Adivinhe 1** - Implemente um programa que sorteia um número de 1 a 10 e dá ao usuário 3 tentativas de acertá-lo. A cada tentativa errada, o programa informa se o número a adivinhar está abaixo ou acima.

**4.2.17 Tabuada** - Elabore um programa que lê um número de 1 a 9 e mostra a tabuada de multiplicação do número. Por exemplo, para o 5:

5 x 1 = 5  
5 x 2 = 10 ...  
5 x 10 = 50

Após mostrar uma tabuada, o programa pergunta se o usuário deseja ver outra. Se a resposta for positiva (por exemplo, 'S'), ele faz tudo de novo, caso contrário termina.

**4.2.20 Conta telefônica** - Uma conta telefônica é composta dos seguintes custos:

assinatura: R\$ 17,90  
impulsos: R\$ 0,04 por impulso que exceder a 90  
interurbanos  
chamadas p/ celular: R\$0,09 por impulso

Elabore um programa que lê número de telefone, valor de interurbanos, quantidade de impulsos normais e para celular, e calcula o valor da conta. Após calcular uma conta, o programa pergunta se o usuário deseja calcular outra conta, reiniciando se a resposta for positiva.

**4.2.23 Série 1** - Escrever programa para calcular, para N lido, o valor de S, dado por:

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

Após efetuar um cálculo, o programa pede novo número, parando quando N for zero.

**4.2.26 Caixa automático** - Um caixa automático precisa calcular quais e quantas notas devem ser entregues ao cliente para efetuar a retirada desejada. Faça um programa com opções para:

- Ler o valor da retirada e mostrar a quantidade de notas de 10 e de 50 a serem entregues. Se alguma das quantidades não for suficiente, o programa cancela a operação, com uma mensagem apropriada. [*Dica para calcular as quantidades de notas: use os operadores **div** e **mod***]
- Receber notas de 10 e 50 (a quantidade inicial é zero para ambas)
- Apresentar relatório com as quantidades de notas e valor total disponível, e valor total de retiradas efetuadas.

## CARACTERES E CADEIAS

**4.3.5 Inserção de caractere 2** - Altere o programa acima para que ele insira um caractere lido do teclado. Ele repete tudo enquanto o usuário digitar uma cadeia. Se nada for digitado, o programa termina. (**4.3.4 Inserção de caractere 1** - Implementar um programa que insere hifens entre as letras de uma cadeia de caracteres, como em f-a-b-u-l-o-s-o.)

**4.3.7 Criptografia 1** - Implementar um programa com duas opções: na primeira, ler e codificar uma cadeia com códigos ASCII de 32 (espaço) a 122 ('z'), da seguinte maneira: ao código ASCII de cada caractere é somado 1; os números resultantes são convertidos novamente em caracteres e concatenados, sendo a cadeia resultante mostrada na tela. A segunda opção decodifica uma cadeia codificada pelo primeiro programa. Exemplo: a cadeia "fogo", codificada, se torna "gphp" (esse processo de codificação é chamado de "criptografia").



**4.3.10 Inversão de cadeia** - Elaborar um programa que lê uma cadeia de caracteres e mostra-a invertida na tela. Após, o programa volta para ler outra, assim fazendo até que a cadeia nula seja nula, isto é, sem nenhum caractere.

## REGISTROS E VETORES

**4.6.7 Pesquisa notas** - Elaborar programa com opções para: ler 10 notas de 0 a 10, pesquisar se uma nota existe no vetor e mostrar o conteúdo do vetor. Na leitura, rejeitar notas fora da faixa válida.

**4.6.9 Validação de senha** - Implementar um programa que lê um nome e uma senha (entre 4 e 8 caracteres) e verifica se o usuário está autorizado ou não. Para essa verificação, o programa mantém uma lista de nomes e respectivas senhas. O programa mostra mensagens de erro se o nome ou a senha estiverem incorretos. São permitidas até 3 tentativas.

**4.6.17 Extenso** - Elaborar um programa que lê um valor monetário e mostra na tela o valor por extenso.

## VARIADOS

**4.8.3 Jogo de fichas** - Altere o programa do sorteio de fichas (3.6.6) para que permita ao jogador jogar quantas vezes quiser. O programa mostra o valor acumulado pelo jogador até um determinado momento. **(3.6.6 Jogo de fichas 2)** - Altere o programa anterior para que leia também o valor apostado, limitado a \$100, e informe o valor a ser recebido pelo apostador.

**4.8.9 CPF** - Os dois dígitos de verificação do CPF (constituído de 9 dígitos) são calculados através de um complicado algoritmo:

### **Etapas 1:** cálculo de DV1

Soma 1: soma dos produtos de cada dígito por um peso de 2 a 10, na ordem inversa (do nono para o primeiro).

Multiplique a soma 1 por 10 e calcule o resto da divisão do resultado por 11. Se der 10, DV1 é zero, caso contrário o DV1 é o próprio resto.

### **Etapas 2:** cálculo de DV2

Soma 2: soma dos produtos de cada dígito por um peso de 3 a 11, também na ordem inversa.

Adicione a Soma 2 ao dobro do DV1, multiplique por 10 e calcule o resto da divisão do resultado por 11. Se der 10, DV2 é zero, caso contrário o DV2 é o próprio resto.

**Etapas 3:** Multiplique DV1 por 10, some com DV2 e você tem o número de controle do CPF.

**Exemplo:** para o CPF 398 136 146, temos:

Etapas 1:  $2 \times 6 + 3 \times 4 + 4 \times 1 + 5 \times 6 + 6 \times 3 + 7 \times 1 + 8 \times 8 + 9 \times 9 + 10 \times 3 = 258$   
 $2580 \text{ mod } 11 = 6$ , portanto, DV1 = 6

Etapas 2:  $3 \times 6 + 4 \times 4 + 5 \times 1 + 6 \times 6 + 7 \times 3 + 8 \times 1 + 9 \times 8 + 10 \times 9 + 11 \times 3 = 299$   
 $(299 + 6 \times 2) \times 10 \text{ mod } 11 = 3150 \text{ mod } 11 = 8$ , portanto DV2 = 8

Etapas 3:  $DV1 \times 10 + DV2 = 6 \times 10 + 8 = 68$ , que é o número procurado.

Elabore um programa que calcule o número de controle do CPF.

**4.8.12 Criptografia 2** - Implementar um programa que criptografa uma cadeia usando o seguinte algoritmo: ela é reescrita em blocos de 5 caracteres, sendo as novas palavras obtidas lendo-se cada coluna resultante, separadas por barras. Por exemplo, se a cadeia for "mensagem secreta":

```
Mensa
gem s
ecret
a
```

A cadeia criptografada torna-se "mgea/eec /nmr/s e /ast. Prever também no programa uma opção para decifrar uma cadeia.

## VARIADOS

**5.7.7 CPF** - Escrever função para calcular os dígitos de controle do CPF (regra na especificação 4.8.9). Para simplificar, já que são duas somatórias, escreva também uma função auxiliar que recebe o CPF e o peso inicial e retorna a soma.

## Arquivos

**4.7.2 Contagem de caracteres** - Implementar um programa que conta a quantidade de caracteres de um arquivo texto.

**4.7.6 Comparação de arquivos** - Faça um programa que compare dois arquivos texto e informe se são iguais (byte a byte) ou, se não, o número do primeiro caractere em que diferem.

**4.7.10 Estatística de texto** - Faça um programa que lê um arquivo texto qualquer e mostra:

```
Quantidade de linhas
Quantidade de letras
Quantidade de palavras
Quantidade de cada letra.
```

**5.6.2 Validação de senha** - Implementar um programa com opções de cadastrar ou autenticar usuário, além de uma opção para terminar. Na opção de cadastro, o programa lê um nome e uma senha (ambos entre 4 e 8 caracteres), sendo esta lida duas vezes, e grava os dados em um arquivo. As duas senhas digitadas devem iguais. Na opção de autenticação, o programa lê nome e senha e verifica se o usuário está cadastrado e se sua senha está correta. O programa mostra mensagens de erro se o nome ou a senha estiverem incorretos, sendo permitidas até 3 tentativas.

**5.6.2 Validação de senha** - Implementar um programa com opções de cadastrar ou autenticar usuário, além de uma opção para terminar. Na opção de cadastro, o programa lê um nome e uma senha (ambos entre 4 e 8 caracteres), sendo esta lida duas vezes, e grava os dados em um arquivo. As duas senhas digitadas devem iguais. Na opção de autenticação, o programa lê nome e senha e verifica se o usuário está cadastrado e se sua senha está correta. O programa mostra mensagens de erro se o nome ou a senha estiverem incorretos, sendo permitidas até 3 tentativas.

**5.6.3 Validação de senha criptografada** - Altere o programa acima de forma que a senha seja gravada criptografada, por exemplo, somando-se 10 ao código ASCII de cada caractere

## 5.6.7 Criptografia de arquivos –

Elaborar um programa que criptografa um arquivo qualquer, incrementando o código ASCII de cada byte em uma unidade.

## Ideias e Desafios

### ENTRETENIMENTO

**6.1.1 Força** - Implementar um programa que jogue o jogo da força. Na tela é mostrado o alfabeto, destacando as letras já tentadas. Um banco de palavras pode ser implementado em vetor ou em arquivos, permitindo ao programa sortear uma palavra. Extensões: armazenar histórico do jogador: nome, jogadas ganhas e perdas etc.

**6.1.2 Palavra embaralhada** - Implementar um programa que, a partir de um banco de palavras, seleciona aleatoriamente uma palavra, embaralha as letras e dá um tempo para o usuário adivinhar a palavra.

### DOMÉSTICOS

**6.2.3 Lista de compras** - Elabore um programa que mantém uma lista de produtos e imprime uma relação para controle da pessoa no supermercado, por exemplo. Inclua uma opção para eliminar itens da lista a ser impressa. Para montar a lista inicial, você pode pesquisar na Internet por um serviço de delivery.

**6.2.5 Controle de empréstimos** - Se você já emprestou um livro ou alguma outra coisa e não se lembra mais para quem (e esse "quem" também parece não se lembrar de devolver), sabe que pode ser útil um programa que registre tipo e nome do objeto, nome da pessoa e as datas de empréstimo, última cobrança e devolução. Inclua um relatório dos objetos emprestados há mais de uma quantidade indicada de dias.

**6.2.6 Orçamento doméstico** - Controle suas entradas e saídas de dinheiro através de um programa que registra despesas e receitas passadas e previstas, fornecendo informações sobre o saldo disponível e previsões de necessidades, mês a mês. Se quiser sofisticar, monte uma tabela de tipos de receitas e despesas (automóvel, lazer, educação etc.).

**6.4.3 Provas V ou F** - Elabore um programa que corrija provas contendo somente questões com respostas Verdadeiro ou Falso. O programa deve armazenar as respostas corretas para cada questão e até 50 provas com até 20 questões. Quando solicitado, o programa calcula e mostra as notas para cada aluno. Opcionalmente, o usuário pode definir um fator de correção, de forma que para cada 2 ou 3 questões erradas, uma certa seja anulada.

**6.5.3 Acha-5** - Dois jogadores, cada um pensa uma palavra de 5 letras diferentes, sem acentos. Um tenta adivinhar primeiro a palavra que o outro escolheu. Alternadamente, cada um submete ao outro uma palavra também de 5 letras, sem regras. O outro vai informar quantas letras daquela palavra estão presentes na sua palavra secreta. Com base nessa informação e, através de lógica, as letras vão sendo eliminadas ou descobertas. Por exemplo, suponha que a palavra secreta do jogador 1 é "zinco". Se o jogador 2 disser "arara", o jogador 1 informa "zero", já que nem o "a" nem o "r" ocorrem em "zinco". O jogador 2 então elimina as duas letras. Se a palavra dita for "ossos", o jogador 1 informa "uma", que pode ser "o" ou "s". Neste ponto (e na sua próxima jogada), o jogador 2 pode dizer "esses" e, ao ser informado que há zero ocorrências, elimina o "e" e o "s" e descobre em "ossos" que a letra "o" pertence à palavra secreta do jogador 1. Implemente esse jogo, com um banco de palavras de 5 letras para que o computador sorteie a sua palavra secreta.

3) Válido para **Programação Modular**:

Como sugestão de autodiagnóstico, considere a seguinte pontuação para os exercícios:

**Básicos:** 2 pontos cada

**Variáveis e entrada de dados:** 4 pontos cada

**Alternativas e decisão:** 6 pontos cada

**Repetição:** 10 pontos cada

**Arquivos:** 10 pontos cada

**Ideias e desafios:** 30 pontos cada

Se você fez o exercício, testou e executou corretamente: 100% da pontuação

Se você tentou fazer o exercício, executou corretamente para alguns casos apenas: 60% da pontuação.

Se você leu o enunciado, tem certeza de que sabe fazer o exercício, mas não fez: 30% da pontuação.

**Obtendo até 60 pontos:** atenção, é preciso reforçar o estudo das habilidades indicadas no exercício o quanto antes.

**Obtendo de 60 a 80 pontos:** você está com uma boa base e provavelmente precisará reforçar alguns conteúdos.

**Obtendo mais de 80 pontos:** parece que você está numa situação bem confortável para o que precisaremos em POO.

4) Válido **para Laboratório de Programação Modular**:

A atividade "Desafio Java – Para Casa" indicará quais dos exercícios você deve entregar obrigatoriamente para avaliação da revisão, com valor de 5 pontos.

Na aula de laboratório da semana de 26 a 29/02 será realizada outra atividade, "Desafio Java – Ao Vivo", na qual você deverá resolver outros exercícios no horário de aula. Esta atividade valerá 10 pontos e será permitida a consulta a materiais de referência para sua realização.