300 exercicios para programar!

Informática

Faculdade Municipal Professor Franco Montoro de Mogi Guaçu (FMPFM)

48 pag.

300 IDÉIAS PARA PROGRAMAR COMPUTADORES

Na minha vivência profissional, como programador, analista de sistemas e professor, tenho visto programadores de vários tipos. Alguns são mais criativos, outros menos, e também há os que parecem eternos iniciantes. Alguns têm sólida formação em algoritmos e técnicas de programação; outros não a tem mas se viram quando precisam. Há programadores que praticamente se "viciam" em comandar computadores, e extraem grande prazer dessa atividade, e há outros que o fazem por fazer, simplesmente, talvez pelo salário.

Os programadores realmente bons, que unem formação técnica, criatividade, curiosidade e prazer parecem ser relativamente poucos. Naqueles que conheci pude perceber padrões: eles têm acesso a um computador, em casa ou em outro lugar; têm uma grande disposição de buscar soluções por si mesmos e, acima de tudo, eles programam muito. Não dependem de alguém mandá-los programar, tendo ou não suas próprias idéias. Para resumir, eles simplesmente **praticam** muito, e este é o fator maior que, na minha opinião, os torna melhores que os outros.

Por isto foi elaborado este material, cujo conteúdo consiste essencialmente de especificações de programas, não voltadas para nenhuma linguagem de programação em particular. Sua maior finalidade é fornecer idéias para que programadores ávidos possam programar, programar e programar, e assim se tornar cada vez melhores. Uma restrição é que a implementação de algumas das especificações será mais apropriada estando a tela em modo texto, o que não impede que muitas delas possam fornecer inspiração para programas em interface gráfica.

A maioria das especificações propostas são voltadas para iniciantes em uma linguagem de programação, e estruturadas na forma que considero mais didática: no início apenas comandos de saída e instruções matemáticas simples, de forma a não sobrecarregar o estudante que, nesse momento, ainda está normalmente assimilando uma série de novas informações sobre processadores, instruções, seqüência, controle e o próprio compilador. Os capítulos seguintes gradativamente incorporam às especificações novos tópicos: variáveis, tipos de dado e entrada, decisão, repetição e outras.

Os primeiros cinco capítulos estão organizados em tópicos, e estes focalizados em estruturas de programação. Os exercícios, embora variados, não são todos diferentes: alguns combinam dois ou mais exercícios de capítulos anteriores, resultando em um mais complexo e no qual o estudante pode aplicar a experiência adquirida.

O último capítulo sugere idéias para programas, existentes ou não, que podem ser implementadas por programadores que já tenham amadurecido os recursos básicos de uma linguagem de programação, podendo também ser usadas para trabalhos práticos de iniciativa pessoal ou determinados pelo professor. Em alguns casos, é dado algum direcionamento inicial para a implementação; em outros, o desafio é todo do programador!

Este é um material para ser usado como um complemento, não servindo por si só para suportar um curso. Dependendo da proposta didática do professor que o adote, será necessário adequar a seqüência em que as especificações serão propostas aos estudantes.

Obviamente não há qualquer limite para a evolução deste trabalho; por isto, críticas, sugestões e eventuais correções serão sempre muito bem vindas.

Virgílio Vasconcelos Vilela

virgilio@tba.com.br

Brasília, Fevereiro/1999.

SUMÁRIO

1.BÁSICOS

Quando começamos, há muita informação nova a ser assimilada. Por isso, convém iniciar pelo mais simples. Para implementar os programas especificados neste capítulo, basta conhecer a estrutura de um programa e algumas poucas instruções (veja o apêndice A). Não são necessárias ainda instruções para tomar decisões ou controlar repetições. Isto lhe dá tempo para assimilar as (possivelmente) novas ferramentas, como editor, compilador e outras.

1.1.SAÍDA SIMPLES

1.1.1 Frase na tela - Implemente um programa que escreve na tela a frase "O primeiro programa a gente nunca esquece!".

1.1.2 Etiqueta - Elabore um programa que, após limpar a tela, escreve seu nome completo na primeira linha, seu endereço na segunda, e o CEP e telefone na terceira.

1.1.3 Frases assassinas - Faça um programa que mostre na tela algumas frases assassinas, que são aquelas que fazem com muitas idéias sejam perdidas antes que amadureçam ou seja aprofundadas. Eis alguns exemplos (bole também os seus):

"Isto não vai dar certo"

"Você nunca vai conseguir"

"Você vai se estrepar"

"Não vai dar em nada"

"Está tudo errado!"

1.1.4 Mensagem - Escreva uma mensagem para uma pessoa de que goste. Implemente um programa que imprima essa mensagem, e envie-a.

1.1.5 Ao mestre - Escreva um bilhete ao seu professor, informando seus objetivos nesta disciplina e o que espera dela e do professor. Implemente um programa que mostra seu bilhete na tela.

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

```
XXXXX
X   X
X   X
X   X
XXXXX
```

1.1.7 Tabela de notas - Escreva um programa que produza a seguinte saída na tela:

```
ALUNO(A)        NOTA
========        =====
ALINE           9.0
MÁRIO           DEZ
SÉRGIO          4.5
SHIRLEY         7.0
```

1.1.8 Apresentação - Um estudante ia participar de uma feira de ciências e seu projeto tinha o tema "fotossíntese". Ele conseguiu um notebook emprestado, e queria um programa que lhe permitisse apresentar um texto dividido em partes, cada parte em uma tela, e o programa deveria mudar para a próxima tela ao toque de uma tecla. A tela inicial deve ser a palavra "FOTOSSÍNTESE" escrita com letras grandes. Faça o programa para o estudante, usando o texto abaixo, no qual cada parágrafo deve aparecer em uma tela diferente. Como o estudante não entende muito de operação de computadores, você tem que também gerar um arquivo executável, copiá-lo para o outro computador e incluir um ícone para fácil execução sob Windows.

Texto:

"A água e os sais minerais absorvidos pelas raízes sobem através dos vasos lenhosos do caule e chegam às folhas.

Nas folhas, existe uma substância verde, a clorofila, que absorve a energia luminosa do sol. Ao mesmo tempo, por meio dos estômatos presentes nas folhas, a planta absorve gás carbônico do ar.

Usando a energia solar, o gás carbônico e o hidrogênio contido na água retirada do solo, após complicadas reações químicas, a planta produz açúcares (glicose)."

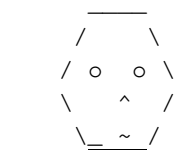
1.1.9 Letra grande - Elabore um programa para produzir na tela a letra X usando a própria. Se fosse 'L', seria assim:

```
L
L
L
L L L L
```

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

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

1.1.11 Desenho datilográfico - Há um artista que faz desenhos somente com caracteres disponíveis em uma máquina de escrever. Bole uma figura nesse estilo (bem melhor que a abaixo, se possível) e faça um programa que a mostre na tela.



1.1.12 Emoticons - *Emoticons* são seqüências de caracteres que mostram rostos e expressões, vistos de lado e usados freqüentemente em correios eletrônicos e bate-papos na Internet. Existem dezenas; veja alguns:

:-) *sorriso*
:-(*tristeza*
:-p *mostrando a língua*
:-o *espanto*
{:-) *cabelo partido ao meio*
:-{ *usa bigode*
:-* *beijo*

Elabore um programa que mostre na tela os emoticons, um em cada linha, com a descrição de cada um.

1.1.13 Pinheiro 1 - Implemente um programa que desenhe um "pinheiro" na tela, similar ao abaixo. Enriqueça o desenho com outros caracteres, simulando enfeites.

```
X
XXX
XXXXX
XXXXXXX
XXXXXXXXX
XXXXXXXXXX
XXXXXXXXXXXX
XXXXXXXXXXXXXX
XX
XX
XXXX
```

1.1.14 Pinheiro 2 -Elabore uma versão 2 do programa do item anterior que desenhe o pinheiro com asteriscos (*). *[Dica: use o recurso de localização/substituição do editor para fazer a substituição rapidamente]*

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.MATEMÁTICA

1.2.1 Expressões aritméticas - Calcule as expressões abaixo, observando a precedência dos operadores. Escreva um programa que mostre na tela o resultado de cada expressão e confira seus cálculos.

2*6 + 11*5 =
20/(-2)/5 =
20/2*2 =
(3+9)/3*4 =
(5*6/(3+2) - 15*4)/6-4 =
4+32*2 -7*2/(9-2) =

1.2.2 Div e mod - Calcule as expressões abaixo; o operador **mod** calcula o resto, e **div**, o quociente da divisão inteira. Depois, escreva um programa que lhe permita verificar, quando executado, se você calculou corretamente.

37 mod 13 =
41 div 7 =
11 div 3 mod 2 =
11 mod 3 div 2 =

1.2.3 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$0,20 por impulso

Monte a fórmula para calcular o valor da conta para 254 impulsos, R$34,29 de interurbanos e 23 chamadas para celular. Elabore um programa que mostra os custos, calcula e mostra o valor total.

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**]*

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.

1.2.6 Média de notas - Monte uma expressão matemática que calcula a média de suas notas (todas) de um período anterior. Faça o cálculo através de um programa, mostrando na tela o resultado, formatado com duas casas decimais e dentro de uma moldura (um retângulo feito com algum caractere).

1.2.7 Conversão de temperatura - Faça um programa que calcula e mostra uma tabela de graus Celsius/Fahrenheit de 1 a 10 [fórmula: C = 5/9(F-32)]. Por enquanto (sem comandos de repetição), você deverá escrever as instruções para calcular e mostrar cada resultado.

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.

1.2.9 Funções matemáticas - Fornecer o valor retornado pelas operações matemáticas abaixo. Depois, chamando as funções adequadas, escreva um programa que lhe permita verificar a correção dos seus cálculos:

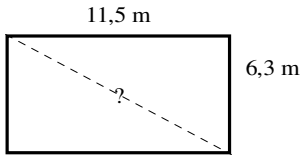
Raiz quadrada de 169

17^2

Cosseno(0)

1.65 arredondado para inteiro

1.2.10 Comprimento de fio - Um eletricista 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.



1.2.11 Tempo dedicado - Uma pessoa com pouco tempo disponível lê um livro por 5 minutos a cada dia, 6 dias por semana. Monte a fórmula e escreva um programa que calcula e mostra na tela quanto tempo, em horas, a pessoa terá dedicado ao livro ao final de um ano.

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.

1.3.CONTROLE DE TELA

1.3.1 Quadrado posicionado - Refaça o programa que desenha o "quadrado" no alto da tela (1.1.6), desta vez desenhando-o com o canto superior esquerdo na linha 7, coluna 20.

1.3.2 Menu posicionado - Elabore um programa que mostre o seguinte menu centralizado na tela, e espera uma tecla ser pressionada para terminar (o traço após "Opção:" é o cursor). Use comandos de posicionamento do cursor para facilitar.

```
          Menu Relatórios

          1 - Por nome
          2 - Por código
          3 - Por data
          4 - Fim

Opção: _
```

1.3.3 Cruz - Elabore um programa que mostra uma linha vertical na coluna 40, formada pelo caractere "#", e uma linha horizontal na posição 10 formada por "=". Entre uma e outra e antes de terminar, o programa espera que uma tecla seja pressionada.

1.3.4 Triângulo com iniciais - Escrever um programa que desenha um triângulo, aproximadamente centralizado, na tela (que em modo texto normal tem 80 colunas por 25 linhas), tendo dentro as iniciais do seu nome. Faça o programa limpar a tela no início e esperar uma tecla antes de terminar.

1.3.5 Apresentação em cores - Altere o programa da fotossíntese (1.1.8) de forma que cada página de texto seja mostrada com uma cor diferente. Destaque palavras específicas do restante do texto com uma cor, como por exemplo, "verde" na cor verde

1.3.6 Animação horizontal - Faça um programa que desenha um "O" na linha 5, coluna 1 e depois faz o seguinte, esperando uma tecla para cada ação (sempre na linha 5):

- apaga o "O" da coluna 1 e o mostra na coluna 2
- apaga da coluna 2 e mostra na 3
- apaga da 3 e mostra na 4

E assim sucessivamente até a coluna 15. Execute o programa mantendo pressionada alguma tecla e veja o resultado.

1.3.7 Animação horizontal - Elabore um programa semelhante ao anterior, mas variando a linha.

1.4.SONS

1.4.1 Afinação - Alberto toca violão e é programador. Precisando afinar o violão e sem diapasão por perto, resolveu fazer um programa para ajudá-lo. O que ele queria era a nota Lá soando sem parar até que ele conseguisse afinar a respectiva corda do violão; as demais cordas ele poderia afinar com base na primeira. Escreva um programa que faz soar no alto-falante do computador a nota Lá (440 Hz) e só para quando for pressionada alguma tecla.

1.4.2 Parabéns - Faça um programa que emite as seis primeiras notas do "Parabéns prá você". Tente as seguintes notas (freqüência em Hz/duração em milissegundos): [440,200], [440,200], [500,800], [440,400], [600,400], [560,800],

2.VARIÁVEIS E ENTRADA DE DADOS

Os programas deste capítulo incorporam, em relação ao anterior, especificações que exigem o uso da memória do computador, incluindo a entrada de dados pelo usuário do programa e o conseqüente armazenamento dos dados lidos.

2.1.SAÍDA SIMPLES

2.1.1 Mensagem emoldurada - Implemente um programa que leia três linhas de mensagens de até 15 caracteres cada uma e mostra-as na tela, emolduradas (retângulo ao redor) por algum caractere.

2.1.2 Etiqueta - Escreva um programa que lê do teclado seu nome completo, endereço, CEP e telefone, limpa a tela e mostra seu nome na primeira linha, seu endereço na segunda, e o CEP e telefone na terceira.

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              x
   xxx            xxx
  xxxxx          xxxxx
 xxxxxxx        xxxxxxx
xxxxxxxxx      xxxxxxxxx
 xxxxxxx        xxxxxxx
  xxxxx          xxxxx
   xxx            xxx
    x              x
```

2.1.4 Losangos 2 - No programa do exercício anterior, troque o caractere de forma que os losangos sejam feitos com asteriscos (*).

2.1.5 Triângulo com iniciais - Escrever um programa que lê um caractere, as iniciais de um nome (3 caracteres), uma linha e uma coluna e depois desenha na tela um triângulo equilátero formado com o caractere, tendo dentro as iniciais lidas. O caractere no ápice do triângulo deve estar na linha e coluna lidas, e a altura do triângulo deve ser no máximo 5 linhas.

2.2.MATEMÁTICA

2.2.1 Média aritmética - Escrever programa que lê três notas inteiras e calcula a sua média aritmética.

2.2.2 Média geométrica - Elabore um programa que lê três valores e calcula a média geométrica dos números lidos (divisão do produto pela quantidade de valores).

2.2.3 Média ponderada - Implemente um programa que lê três valores e calcule a média ponderada para pesos 1, 2 e 3, respectivamente (multiplique cada nota pelo seu peso, some os produtos e divida o resultado pela soma dos pesos).

2.2.4 Aritmética básica - Implemente um programa que lê dois números quaisquer e informa sua soma, diferença, produto e quociente, formatados com 2 casas decimais.

2.2.5 Funções matemáticas - Elabore um programa que lê um número (suponha que será positivo) e informa seu quadrado, raiz, logaritmo e exponencial, formatados com 4 casas decimais

2.2.6 Nota final - O critério de notas de uma faculdade consiste de uma nota de 0 a 10 em cada bimestre, sendo a primeira nota peso 2 e a segunda peso 3. Elabore um programa que lê as notas bimestrais e calcula a nota do semestre.

2.2.7 Soma das idades - Uma criança quer saber qual é a soma de todas as idades que ela já teve. Elaborar programa que lê uma idade qualquer e responde rapidamente a essa pergunta *[fórmula para calcular a soma dos N primeiros números inteiros: N (N+1)/2]*.

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.

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

2.2.10 Conversão de temperatura - Um canal de notícias internacionais, a cabo, previa temperatura máxima para Brasília de 85 graus Fahreneit. Escrever um programa que lhe permita converter esta temperatura (e qualquer outra) para graus Celsius, sabendo que a relação entre elas é C =5/9 (F - 32).

2.2.11 Quantidade de flexões - Um atleta faz flexões em série, com quantidades crescentes: 1 vez, depois 2 vezes, 3, 4 e assim por diante. Ao final de uma sessão, ele quer saber rapidamente a quantidade total de flexões que fez. Por exemplo, se ele fez 5 seqüências, fez ao todo 15 flexões (5+4+3+2+1). Implemente um programa que leia o número máximo e informe o total.

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
```

2.2.13 Despesas de casal 2 - Altere o programa acima de forma que o marido arque com 60% das despesas e a esposa com o restante.

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 seqüência 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*].

2.2.16 Conversão cm/pol 1 - Faça um programa que mostra 10 linhas de uma tabela de conversão centímetro/polegada, a partir de um valor lido e variando de 10 em 10 centímetros (uma polegada equivale a 2,54 centímetros).

2.2.17 Conversão cm/pol 2 - Altere o programa do exercício anterior de forma que a variação também seja lida do teclado.

2.2.18 Otimização de corte - Reescreva o programa 1.2.5 (corte de tábuas) para que leia o tamanho de cada tábua e o comprimento de cada pedaço, e calcule a quantidade de pedaços e a sobra para cada tipo de tábua.

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 cada 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

2.3.1 Concatenação - Escreva um programa que lê duas cadeias de caracteres de tamanho 10 e mostra-as concatenadas na tela.

2.3.2 Subcadeias - Escreva um programa que lê uma cadeia de caracteres de tamanho 20, separa-a em duas e mostra na tela as duas metades.

2.3.3 Códigos ASCII - Escreva um programa que lê uma cadeia de caracteres qualquer, e mostra na tela o código ASCII do primeiro e segundo caracteres da cadeia.

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

2.3.5 Finais - Reescreva o programa anterior para mostrar na tela as letras finais do nome e sobrenome.

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]*

2.3.7 Códigos ASCII inicial e final - Elabore um programa que lê um nome de até 15 caracteres e mostra a inicial e seu código ASCII, e a última letra e seu código.

2.3.8 Soma de códigos ASCII - Escreva um programa que lê uma cadeia de tamanho 3 e mostra na tela a soma dos códigos ASCII dos caracteres 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.

2.4.CONTROLE DE TELA

2.4.1 Animação horizontal - Faça um programa que lê valores de linha e coluna e desenha um "O" na posição lida, e depois faz o seguinte, esperando uma tecla para cada ação (sempre na mesma linha):

- apaga o 'O' da posição atual
- incrementa a coluna
- mostra o 'O' na nova posição

E assim sucessivamente por 10 colunas.

2.4.2 Quadrado em posição - Escrever um programa que desenha um quadrado com o canto superior esquerdo em uma linha e coluna lidas. O caractere usado para formar o quadrado é o '#'. Veja abaixo uma sugestão para a tela do programa.

```
Este programa desenha um quadrado com o caractere #
Linha: 10
Coluna: 30









                              #####
                              #   #
                              #   #
                              #   #
                              #####
Pressione qualquer tecla
```

2.4.3 Triângulo com iniciais - Faça um programa que lê valores de linha e coluna, além das iniciais de um nome (até 3 caracteres) e desenha um triângulo ("bole" o desenho) com um vértice na linha e coluna lidas e com as iniciais dentro.

2.4.4 Menu posicionado - Implemente um programa que mostra o menu abaixo a partir de uma linha lida do teclado:

```
                    Menu de Consultas

                    0 - Fim
                    1 - Clientes
                    2 - Produtos
                    3 - Faturas
                    4 - Estoque

Opção: _
```

2.5.SONS

2.5.1 Nota musical - Elaborar um programa que lê uma freqüência (em Hertz) e uma duração (em milissegundos) e emite um som na freqüência com a duração.

2.5.2 Música é linear? - A respeito do programa do parabéns (1.4.2), deseja-se saber se a melodia é preservada somando-se um valor constante a cada freqüência. Faça um programa que lê essa constante (por exemplo, 100) e faz soar as notas somando a constante a cada freqüência.

3.ALTERNATIVAS E DECISÃO

O recurso principal incorporado nas especificações deste capítulo é a possibilidade de executar condicionalmente um conjunto de instruções. Uma ou mais condições, na forma de expressões lógicas, são avaliadas, para determinar o que será executado. Algumas especificações são semelhantes às de capítulos anteriores, acrescidas de alternativas ou validações.

3.1.SAÍDA SIMPLES

3.1.1 Menu principal - Faça um programa de menu que mostra na tela, sob o título de "Menu Principal", três opções: "1 - Fim", "2 - Cadastro" e "3 - Consulta", lê do teclado a opção desejada pelo usuário e mostra uma mensagem confirmando a opção escolhida ou uma mensagem de erro, se a opção for inválida.

3.1.2 Múltipla escolha 1 - Elaborar uma questão de múltipla escolha, de uma disciplina que esteja cursando ou um tema de interesse, com um enunciado e cinco alternativas, sendo uma correta ou incorreta. Escrever um programa que mostra a questão na tela, pede a resposta correta e informa ao usuário se este acertou ou errou.

3.1.3 Múltipla escolha 2 - Enriqueça o programa acima da questão de múltipla escolha, incluindo uma outra questão de outro tema. No início do programa, ofereça ao usuário a escolha de qual questão quer responder.

3.2.MATEMÁTICA

3.2.1 Maior de 2 - Elaborar programa que lê dois números quaisquer e mostra na tela uma mensagem indicando qual é o maior, ou se são iguais.

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.

3.2.3 Divisão - Escrever um programa que lê dois números e efetua uma divisão, mas somente se o divisor for diferente de zero; quando isto ocorrer, é mostrada uma mensagem de erro apropriada.

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.

3.2.5 Aprovação 2 - Altere o programa acima para que, se a nota estiver fora da faixa válida, seja emitida uma mensagem de erro.

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 freqüência, ele está reprovado, independentemente da nota.

3.2.7 Equação do segundo grau - Elaborar programa que lê os coeficientes a, b e c de uma equação de segundo grau e, antes de calcular as raízes, calcula o delta. Se este for negativo, informa que a equação não tem solução real. Se for zero, mostra a única raiz. Se positivo, mostra as duas raízes.

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.

3.2.11 Notas do professor - Reescreva o programa 2.2.19 para que, caso uma das notas esteja fora da faixa válida, o programa mostre uma mensagem de erro e não efetue o cálculo.

3.2.12 Menção - Uma faculdade atribui menções aos alunos conforme a faixa de notas que tenha atingido:

9,0 a 10:	SS (superior)
7,0 a 8,9:	MS (médio superior)
5, 0 a 6,9:	MM (médio)
3,0 a 4,9:	MI (médio inferior)
0,1 a 2,9:	II (inferior)
0 :	SR (sem rendimento).

Faça um programa que lê a nota e informa a menção.

3.2.13 Notas finais - As notas de uma faculdade são atribuídas por bimestre, tendo o primeiro bimestre peso 2 e o segundo peso 3. A nota semestral deve ser arredondada para o múltiplo de 0,5 mais próximo. Elabore um programa que calcule a nota final.*[Dica para o arredondamento: obtenha as partes inteira e fracionária da nota; com base na fração, decida se soma 0, 0,5 ou 1 à parte inteira]*

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.

3.2.15 Ano bissexto - Um ano é bissexto se for divisível por 4 exceto os séculos, que são bissextos se forem múltiplos de 400. Escreva um programa que determina se um ano é bissexto.

3.3.CARACTERES E CADEIAS

3.3.1 Tipo de pessoa - Elaborar programa que lê do teclado uma letra que pode ser 'F' ou 'J' e mostra a mensagem "pessoa física", "pessoa jurídica" ou "tipo de pessoa inválido", conforme o caso.

3.3.2 Caracteres ASCII - Escreva um programa que lê três números de 32 a 254 e mostra na tela uma cadeia formada pela concatenação dos caracteres ASCII de cada número. Caso algum dos números esteja fora da faixa válida, o programa mostra uma mensagem de erro apropriada.

3.3.3 Validação de senha - Elabore um programa que lê uma senha de até 8 caracteres, verifica se a senha está correta ou não, comparando-a com uma senha predefinida, e informa "Acesso autorizado" ou "Acesso negado", conforme o caso.

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.

3.3.5 Código ou caractere ASCII - Escreva um programa que lê uma opção que pode ser 1 ou 2. Se o usuário escolher 1, o programa lê um número de 1 a 255 e mostra o caractere ASCII correspondente; se 2, é lido um caractere e mostrado o respectivo código ASCII. Criticar as entradas numéricas e mostrar mensagens apropriadas em caso de erro.

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

3.3.7 Sorteio da LBV - Reescreva o programa 2.3.10 para que verifique se os números lidos estão no formato esperado (por exemplo, 21.375). Caso algum esteja incorreto, o programa mostra uma mensagem de erro.

3.4.CONTROLE DE TELA

3.4.1 Quadrado posicionado - Elabore um programa que mostre um "quadrado" de lado 5 na tela, a partir de uma linha e uma coluna lidas do teclado. Se algum dos valores estiver fora da faixa válida, é mostrada uma mensagem de erro e o desenho não é mostrado.

3.4.2 Quadrado ou triângulo - Implemente um programa com 3 opções (letra ou número): terminar, desenhar um quadrado ou um triângulo na tela, em linha e coluna lidas pelo teclado. Elabore o quadrado e o triângulo como achar melhor. Faça o programa mostrar uma mensagem de erro se o usuário escolher uma opção inválida ou informar valor inválido para linha ou coluna.

3.5.SONS

3.5.1 Nota musical - Elaborar um programa que lê uma freqüência em Hertz e uma duração em milissegundos e emite um som na freqüência com a duração. Limite a freqüência até 10.000 Hz e a duração a 2 segundos.

3.6.VARIADOS

3.6.1 Cadeia centralizada - Elabore um programa que lê um número de linha e uma cadeia qualquer, limpa a tela e mostra a cadeia centralizada na linha indicada. Linhas inválidas não são aceitas. *[Dica: calcule a coluna com base na quantidade de colunas da tela e no comprimento da cadeia]*

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:

1x3 + 7x2 + 0x9 + 1x8 + 2x7 + 0x6 + 9x5 + 0x4 + 4x3 + 1x2 = 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:

Primeira Ficha	*Segunda Ficha*	*Rateio*
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.

3.6.6 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.

3.6.7 Jogo de fichas 3 - Modifique o programa do jogo de retirada de fichas, acima, de forma que o jogador retire três fichas. Atribua valores de rateio para todas as combinações de cores . Implemente um programa que lê o valor apostado, sorteia as cores, calcula o rateio obtido pelo jogador e o valor que ele receberá.

3.6.8 Adivinhe 1 - Faça um programa que sorteia um número de 1 a 5 e pede ao usuário que o adivinhe, lendo do teclado o palpite. Caso o usuário acerte ou não, é mostrada uma mensagem apropriada.

3.6.9 Adivinhe 2 - Modifique o programa acima para que o usuário possa tentar novamente se errar na primeira vez.

3.6.10 Categoria de altura - Elaborar programa que lê uma altura e mostra uma mensagem conforme a faixa de altura:

menos que 1,60	"baixinho"
de 1,60 a 1,85	"altura normal"
mais que 1,85	"faz frio aí em cima?"

3.6.11 Conceito - Uma universidade atribui conceitos aos alunos com base na nota obtida em cada disciplina, segundo a tabela abaixo. Escreva um programa que lê a nota e informa o conceito obtido.

NOTA	CONCEITO
90..100	A
75..89	B
50..74	C
40..49	D
0..39	E

3.6.12 Multiplicação rápida - Um algoritmo para multiplicação rápida por 11 de números de 2 dígitos funciona assim: para multiplicar 81 x 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 x 11 = 616. Faça um programa que efetue multiplicações por 11 usando este algoritmo.

4.REPETIÇÃO

É difícil imaginar um programa realmente útil que não contenha comandos de repetição. Familiarize-se com todos os tipos de comandos de repetição disponíveis na linguagem, para que possa usar o melhor para cada situação.

4.1.SAÍDA SIMPLES

4.1.1 Egocentrismo - Implemente um programa que mostra seu nome na tela dez vezes.

4.1.2 Mais pinheiro - Fazer um programa que desenha o pinheiro (1.1.13), usando comandos de repetição.

4.1.3 Caracteres progressivos 1 - Escrever um programa que produza a saída abaixo na tela, para N linhas e usando um caractere lido do teclado (no exemplo, *). Após mostrar uma vez, o programa repete o processo, só parando quando N for zero.

```
**
****
******
********
(...)
```

4.1.4 Caracteres progressivos 2 - Faça o mesmo que acima para:

```
*
**
***
****
*****
(...)
```

4.1.5 Caracteres progressivos 3 - Idem acima, para o formato abaixo.

```
**
****
******
********
(...)
```

4.1.6 Caracteres progressivos 4 - Ibidem:

```
(...)
*****
***
*
```

4.1.7 Tudo junto - Faça um programa que junte os 4 exercícios acima. Ele repetidamente oferece um menu com a opção 0 para terminar e outras 4 opções 1, 2, 3,e 4, cada uma correspondendo a um tipo de figura. Caso a opção indicada pelo usuário seja inválida, é mostrada uma mensagem apropriada. Em todos os casos exceto 0 o menu é oferecido novamente. Tente estruturar o

programa de forma que a leitura da quantidade de linhas seja feita em apenas um ponto do programa, ao invés de ser lida a cada opção.

4.2.MATEMÁTICA

4.2.1 Aprovação - Elaborar programa que lê uma disciplina e respectiva nota (de 0 a 10, com uma casa decimal), e informa se o aluno passou na disciplina, repetindo o ciclo até que a nota lida seja zero. O aluno passa quando tira 7 ou mais.

4.2.2 Raiz quadrada 1 - Implemente um programa que repetidamente calcula e mostra a raiz quadrada de um número qualquer.

4.2.3 Raiz quadrada 2 - Altere o programa acima para que ele verifique se o usuário entrou um valor positivo ou zero. Se sim, a raiz é calculada, caso contrário é mostrada uma mensagem de erro.

4.2.4 Idade média - Um professor, após obter informações de uma turma, deseja saber a média de idade. Escrever um programa que lê as idades até que o idade lida seja zero, quando então é mostrada a média (o zero não é considerado para a média).

4.2.5 Estatística de notas - Faça um programa que lê uma quantidade qualquer de notas de 0 a 10 (não permitir fora desta faixa) e, ao final, mostra quantas notas foram digitadas, a média e também a quantidade com valor abaixo de 5 .

4.2.6 Maior - Escrever um programa que lê números inteiros até que o número lido seja zero, quando então é mostrado o maior número lido.

4.2.7 Maior e menor - Alterar o programa anterior para que mostre também o menor número lido.

4.2.8 Números inteiros 1 - Escrever um programa que lê um número inteiro e mostra na tela os números inteiros de 1 até o número lido.

4.2.9 Números inteiros 2 - Alterar o programa acima de forma que seja lido também o número inicial.

4.2.10 Soma de pares - Implemente um programa que calcula a soma dos números pares compreendidos entre dois números lidos.

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.

4.2.13 Conversão de temperatura 1 - Escrever um programa que mostra uma tabela de graus Celsius/Fahrenheit de 0 a 100, variando 1 grau de cada vez, uma temperatura por linha. Ao encher uma tela, o programa espera que uma tecla seja pressionada para continuar.

4.2.14 Conversão de temperatura 2 - Alterar o programa acima de forma que sejam lidas do teclado a temperatura inicial, a final e a variação. A temperatura final é criticada; se for menor do que a inicial, o programa repete a leitura, só prosseguindo quando for válida.

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.

4.2.16 Adivinhe 2 - Altere o programa acima para que ele permita ao usuário tentar até acertar.

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.

4.2.18 Raiz quadrada - Faça um programa que repetidamente mostra na tela duas opções: "1 - Fim" e "2 - Calcular raiz" e lê do teclado a opção desejada pelo usuário. Se a opção for 1, o programa termina. Se a opção for 2, o programa lê um número real e, se o número for positivo ou zero, calcula e mostra sua raiz quadrada com duas casas decimais, se negativo, mostra uma mensagem de erro. E se a opção for inválida (nem 1 nem 2), é mostrada uma mensagem apropriada. Quando a opção não é a de terminar, o programa volta para mostrar novamente as opções e ler a opção do usuário.

4.2.19 Numerador - Implemente um programa que mostre na tela os números inteiros entre dois números lidos do teclado, organizados em 10 linhas e 10 colunas:

21 22 23 24 25 26 27 28 29 30
31 32 33 34...

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.

4.2.21 Contas telefônicas - Faça uma versão do programa acima que mostre a quantidade de contas, o valor total e a média do valor das contas, quando o usuário terminar.

4.2.22 Eleição - Para computar o vencedor de uma eleição deve ser feito um programa. Há 3 candidatos, e os votos dos eleitores foram codificados da seguinte forma:

1, 2 ou 3: votos para os respectivos candidatos
0: voto em branco
4: voto nulo

Escrever o programa, que deve fornecer o número do vencedor da eleição (suponha que não pode haver empates), as quantidades de votos brancos e nulos e o número de eleitores que compareceram às urnas.

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

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

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

4.2.24 Série 2 - Sendo $H 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{N}$, elaborar um programa para calcular o valor de H, para N lido.

4.2.25 Série convergente - A série $S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \Lambda$ converge para 1. Demonstre isso através de um programa que calcula o valor de S para N termos. Para facilitar, após apresentar um resultado, faça o programa voltar para ler outro valor de N, só parando quando o número lido for zero. Execute várias vezes para valores sucessivamente maiores de N, e observe os resultados.

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:

a) 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**]*

b) Receber notas de 10 e 50 (a quantidade inicial é zero para ambas)

c) Apresentar relatório com as quantidades de notas e valor total disponível, e valor total de retiradas efetuadas.

4.3.CARACTERES E CADEIAS

4.3.1 Sorteio da LBV - Melhore o programa 3.3.7 de forma que, quando houver um erro na digitação, ele permita ao usuário entrar novamente, só continuando quando os dois números estiverem corretos.

4.3.2 Soma de códigos ASCII - Escreva um programa que lê uma cadeia de caracteres quaisquer e mostra a soma dos códigos ASCII dos seus caracteres. Isto é repetido até que a cadeia lida seja nula.

4.3.3 Concatenação de caracteres - Elabore um programa que lê uma quantidade qualquer de números de 32 a 254 e mostra na tela uma cadeia formada pela concatenação dos caracteres ASCII de cada número. Se um dos números estiver fora de faixa, é mostrada uma mensagem de erro e o programa espera a correção. O final da entrada de números ocorre quando for lido zero.

4.3.4 Inserção de caractere 1 - Implementar um programa que insere hífens entre as letras de uma cadeia de caracteres, como em f-a-b-u-l-o-s-o.

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.

4.3.6 Substituição de caractere - Elabore um programa que troca todos os espaços de uma cadeia lida por um caractere também lido. O programa repete isso até que seja lida uma cadeia nula (neste caso o caractere não é lido).

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 pela primeira programa. Exemplo: a cadeia "fogo", codificada, se torna "gphp" (esse processo de codificação é chamado de "criptografia").

4.3.8 Prenome - Escrever um programa que lê um nome completo e mostra na tela o prenome, isto é, o primeiro nome. Suponha que o nome nunca começa com um espaço. O programa repete esses passos até que o nome lido seja uma cadeia nula (o usuário não digitou nada).

4.3.9 Iniciais - Escreva um programa que lê um nome de pessoa e identifica suas iniciais, segundo o seguinte critério: uma inicial é o primeiro caractere ou o caractere que segue um espaço.

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.

4.3.11 Eliminação de caractere - Fazer um programa que lê uma cadeia e um caractere e elimina todas as ocorrências do caractere na cadeia. Após, o programa pede nova cadeia e só termina quando a cadeia lida for nula.

4.3.12 Quantidade de caracteres - Elaborar um programa que lê uma cadeia e um caractere e informa a quantidade de ocorrências do caractere na cadeia (não diferenciar minúsculas/maiúsculas: 'a' = 'A'). Por exemplo, se a cadeia for "BANANA nanica" e o caractere for "a", o programa deve informar 5.

4.3.13 Estatística de frase 1 - Elabore um programa que lê uma cadeia de até 255 caracteres e informa:

- quantidade de brancos
- quantidade de palavras
- quantidade de ocorrências da letra 'A'

4.3.14 Estatística de frase 2 - Altere o programa acima para que informe também a quantidade de cada vogal.

4.3.15 Validação de senha - Escrever um programa que lê uma senha (entre 4 e 8 caracteres), compara a senha linha com o valor correto e informa se o usuário está autorizado ou se a senha está incorreta. A senha correta é registrada dentro do programa como uma constante. O programa permite até 3 tentativas.

4.4.CONTROLE DE TELA

4.4.1 Animação horizontal 1 - Implementar um programa que simula um caractere se movendo pela tela ao longo de uma linha cujo valor é lido do teclado, a partir e até colunas também lidas. O programa verifica se a coluna final informada é maior do que o valor inicial.

4.4.2 Animação horizontal 2 - Alterar o programa anterior para que o movimento seja na vertical, isto é, a coluna fica fixa e a linha varia.

4.4.3 Nave espacial 1 - Bolar um desenho de uma "nave espacial" em modo texto. Fazer um programa que inicia com a "nave" no centro da tela e move-a para a esquerda ou direita se tecladas as setas, terminando se teclado ESCAPE. A nave pára quando atinge os limites da tela.

4.4.4 Nave espacial 2 - Alterar o programa acima para permitir o movimento também para cima e para baixo.

4.4.5 Desenho - Faça um programa de desenho, cuja tela tem um cursor que se movimenta com as setas, deixando um "rastro" (algum caractere; se quiser sofisticar, ponha um hífen se o movimento for na horizontal e uma barra, se na vertical. Se quiser sofisticar mais ainda, use os caracteres de desenho de retângulos da tabela ASCII).

4.4.6 Apresentação 1 - Altere o programa da fotossíntese (1.1.8) de forma que o estudante avance ou retroceda as páginas através de seta acima e seta abaixo. O programa termina quando avançar além da última página ou quando teclado ESCAPE. *[Dica: use um comando de repetição, combinado com dois comandos de decisão. O primeiro, após a leitura da tecla, ajusta a página a ser mostrada. O segundo mostra o texto da página]*.

4.4.7 Apresentação 2 - Altere o programa anterior de forma que, quando o usuário teclar o número de uma página existente, o programa vai direto para a página correspondente.

4.4.8 Losangos - Refaça o programa dos losangos (2.1.3), desta vez usando comandos de repetição.

4.4.9 Quadrados crescentes - Elaborar um programa que mostra um "quadrado" no centro da tela, de lado 2. Em seguida, mostra outro quadrado de lado 4 ao redor do primeiro e apaga este, depois um de lado 6, etc., até "sair" da tela. O programa repete isso até que uma tecla seja pressionada.

4.5.SONS

4.5.1 Chateação - Implementar um programa que fica repetindo a melodia do parabéns (especificação 1.4.2) até que uma tecla seja pressionada. Execute-o sempre que quiser amolar alguém!

4.5.2 Som crescente 1 - Elaborar um programa que emite sons de freqüência crescente, iniciando em 100 Hz até cerca de 8000 Hz, com variação de 10 em 10 % e tendo cada som a duração de 30 milissegundos.

4.5.3 Som crescente 2 - Alterar o programa acima para que leia via teclado todos os valores: freqüências inicial e final, duração do som e variação. Verificar se a freqüência final é maior do que a inicial e se a variação é maior do que 1 e menor do que 2.

4.5.4 Som decrescente - Reescrever o programa acima para que a emissão de sons seja com freqüência decrescente.

4.5.5 Queda - Implementar um programa que emite sons de 700 a 600 Hertz, variando a frequência de 1 em 1, tendo cada som a duração de 10 milissegundos.

4.6.REGISTROS E VETORES

4.6.1 Média - Escrever um programa que leia até 20 números inteiros para um vetor e calcule a média dos valores.

4.6.2 Maior e menor - Escrever um programa que preencha um vetor de 100 elementos com valores inteiros aleatórios, e identifique o maior e o menor número gerados, e respectivas posições.

4.6.3 Média ponderada - Escrever um programa que calcula médias ponderadas para uma quantidade de fatores de até 15. O programa lê vários pares [número, peso] até que seja lido um número negativo. É calculada então a média, somando-se os produtos de cada número por seu peso e dividindo-se o resultado pela soma dos pesos.

4.6.4 Soma em vetor 1 - Escrever um programa que, após preencher dois vetores com números inteiros aleatórios, soma os valores dos elementos correspondentes de dois vetores, armazenando o resultado num terceiro vetor.

4.6.5 Soma em vetor 2 - Elabore um programa semelhante ao anterior, exceto que, em apenas um vetor, soma os valores de dois campos e armazena o resultado em um terceiro campo de um registro.

4.6.6 Tabela de temperatura - Implementar um programa que monta uma tabela de graus Celsius/Fahrenheit desde o ponto de fusão até o ponto de ebulição da água, em incrementos unitários. Após são oferecidas opções para o usuário ver na tela ou imprimir.

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.

4.6.8 Nome do dia - 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 estiver fora da faixa válida, é mostrada uma mensagem de erro.

4.6.9 Validação de senha - Implementar um programa que lê um nome e uma senha (entre 4 e 8 caracteres) e verifica e 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.

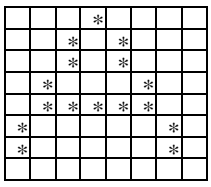
4.6.10 Alunos e notas - Implemente um programa que lê uma lista de pares nomes de aluno/notas. Depois são mostrados na tela os nomes e as notas, juntamente com a quantidade de alunos e a média das notas.

4.6.11 Troco - Implemente um programa que resolve o problema do troco: dado um valor de uma venda, um valor pago e um estoque de notas e moedas (todos os possíveis), calcular o troco e as notas ou moedas e respectivas quantidades que devem ser entregues ao cliente. Procure estruturas de dados que permitam soluções mais simples, como por exemplo um vetor com o valor de cada nota ou moeda, em ordem decrescente de valor.

4.6.12 Frases espelhadas - Faça um programa que leia cadeias (qualquer quantidade, limitada a 20) de até 39 caracteres e mostre-as espelhadas no centro da tela, como no exemplo:

```
Primeira ariemirP
 Segunda adnugeS
Terceira ariecreT
       ...
```

4.6.13 Palavras grandes - Faça um programa que lê uma cadeia de até 10 caracteres e a mostra na tela com letras grandes. Cada letra é formada por uma matriz 8x8, com algum caractere nas posições adequadas de forma a compor o desenho de cada letra (cada caractere é como se fosse um pixel - veja sugestão abaixo). Para separar uma letra da outra, quando mostradas na tela, você pode deixar em branco uma linha e uma coluna de cada letra, na própria matriz.



4.6.14 Rifa - Uma rifa é sorteada com base nos números da Loteria Federal da seguinte maneira: o primeiro prêmio é formado obtendo-se o primeiro dígito de cada prêmio. O segundo é obtido através dos segundos dígitos, e assim por diante. Por exemplo, suponha que os números da Loteria Federal são:

1 - **4**5.698
2 - **6**5.788
3 - **0**1.214
4 - **3**7.840
5 - **7**7.430

Os prêmios da rifa serão 46.037, 55.177, etc.

Escreva um programa que lê os números da Loteria Federal e calcula os números da rifa [*Dica: armazene os números como cadeias em um vetor*]

4.6.15 Sena - Faça um programa que lê apostas da sena, os números sorteados e apresente os resultados obtidos pelo apostador: senas, quinas e quadras.

4.6.16 Codificação de preço - Certas lojas usam (ou pelo menos usavam) um sistema de codificação de preços associando os dez dígitos à palavra PERNAMBUCO. Implemente um programa que lê um preço e mostra-o codificado, ou lê um código e mostra o preço.

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

4.7.ARQUIVOS

4.7.1 Cópia backup - Escrever um programa que lê um nome de arquivo texto existente e copia-o para um outro com extensão ".bak". Caso o arquivo não seja encontrado, é mostrada uma mensagem de erro.

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

4.7.3 TYPE 1 - Elaborar um programa que lê um arquivo texto qualquer e o mostra na tela. Inclua opções para converter as letras para maiúsculas e para filtrar (não mostrar) os caracteres ASCII com código abaixo de 32 ou acima de 127..

4.7.4 TYPE 2 - Altere o programa acima para que numere as linhas ao mostrá-las.

4.7.5 Contagem de palavras - Implementar um programa que conte as palavras contidas em 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.

4.7.7 Cadastro de notas - Fazer um programa que armazena arquivos contendo nome e notas bimestrais de alunos. O nome do arquivo identifica a disciplina. Para que o programa identifique quais arquivos são dele, defina uma extensão padrão para o nome, como por exemplo, ".dat". Declare opções para incluir, alterar e excluir aluno. Preveja também um relatório completo de uma disciplina, contendo, além do nome e das notas parciais, a nota final.

4.7.8 Contagem de linhas de código 1 - Escrever um programa que conta linhas de código de um programa da linguagem em que estiver programando. Mostrar na tela a quantidade total de linhas, linhas em branco e o saldo.

4.7.9 Contagem de linhas de código 2 - Alterar o programa acima para que mostre e desconte a quantidade de linhas de comentários (o programa terá que identificar o início e o fim de cada comentário).

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.

4.7.11 Impressão - Faça um programa que imprime um arquivo texto. Antes de imprimir, pergunta ao usuário a quantidade de linhas por página, se deseja ou não numeração das páginas e ainda permite a digitação de um cabeçalho que, se fornecido, será impresso na mesma linha da numeração de páginas.

4.8. VARIADOS

4.8.1 Parabéns - Para o programa do Parabéns (pág. 13), armazenar as notas e durações em um vetor e reproduzir a melodia a partir do vetor.

4.8.2 Melodia - Elaborar um programa que lê uma seqüência de várias notas definidas por pares freqüência (Hz)/duração (milissegundos), armazena-os em um vetor e "toca" a melodia.

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.

4.8.4 Tela aleatória - Escreva um programa que fica preenchendo a tela com caracteres ASCII aleatórios, em uma posição também aleatória. Quando atingir 1000 caracteres, a tela é limpa e tudo recomeça, só parando quando alguma tecla for pressionada.

4.8.5 Cheques - Faça um programa para preencher cheques. A data de emissão é lida na forma "DD/MM/AA". O programa separa dia, mês e ano e preenche o nome do mês.

4.8.6 Sorteio de consórcio - Um consórcio sorteia seus carros com base na Loteria Federal da seguinte maneira: o premiado de um grupo é o que tiver a pedra correspondente à dezena final do primeiro prêmio. Se ele já tiver sido contemplado, a próxima dezena é formada pelo dígito do milhar e da dezena. Se o consorciado desta pedra também já foi contemplado, a nova dezena inclui o milhar e a centena, e assim por diante, pegando-se até 3 dezenas de cada número. Por exemplo, suponha que os números da Loteria Federal são:

1 - 45.6**98**

2 - 65.788

3 - 01.214

4 - 37.840

5 - 77.430

As dezenas consideradas serão 98, 69, 56 (primeiro prêmio), 88, 78, 57 (segundo prêmio), e assim por diante.

Faça um programa que lê os cinco prêmios e mostra as dezenas sorteadas, na ordem correta.

4.8.7 Impressão de programa fonte - Faça um programa que imprime um arquivo contendo um programa fonte da linguagem que usa. Na impressão são ressaltadas em negrito as palavras chave da linguagem, e os comentários são impressos em itálico. Ao final o programa mostra a quantidade de linhas somente de comentários, linhas em branco e total de linhas.

4.8.8 Linha reta - Implemente um programa que dê uma inclinação e desenha na tela, ponto a ponto, uma linha reta com a inclinação lida. Uma reta é definida pela equação $y = ax + b$, onde a é a inclinação.

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

Etapa 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.

Etapa 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.

Etapa 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:

Etapa 1: 2x6 + 3x4 + 4x1 + 5x6 + 6x3 + 7x1 + 8x8 + 9x9 + 10x3 = 258

2580 mod 11 = 6, portanto, DV1 = 6

Etapa 2: 3x6 + 4x4 + 5x1 + 6x6 + 7x3 + 8x1 + 9x8 + 10x9 + 11x3 = 299

(299 + 6x2)x10 mod 11 = 3150 mod 11 = 8, portanto DV2 = 8

Etapa 3: DV1x10 + DV2 = 6x10 + 8 = 68, que é o número procurado.

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

4.8.10 Senha invisível - Modifique o programa de validação de senha (0) para que ,quando a senha for digitada, não seja visível, impedindo que alguém a descubra.

4.8.11 Apresentação - Modifique a versão mais elaborada do programa da fotossíntese (4.4.6) de forma que as páginas seja armazenadas em vetor. Descubra uma boa estrutura de dados para simplificar o problema.

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.

5.CRIAÇÃO DE INSTRUÇÕES

Há dois tipos básicos de instruções que podem ser declaradas pelo programador: procedimentos e funções. A principal diferença entre elas é que a função retorna um valor. Poder declarar instruções, estendendo a linguagem, certamente é o recurso mais valioso das linguagens de programação, por proporcionar reusabilidade e portanto produtividade. É conveniente também saber como armazená-las em bibliotecas, para que elas sejam *facilmente* reutilizáveis.

Mas lembre-se: uma instrução só será reutilizável se puder ser usada sem qualquer outra declaração, isto é, é preciso saber apenas o seu nome e seus parâmetros. Também não será conveniente que a instrução mostre resultados na tela, a menos que isso faça parte da sua finalidade. Por exemplo, se a finalidade for "calcular a média", a instrução retorna o valor calculado mas não mostra na tela. Se a finalidade for "Mostrar uma mensagem na tela", então ela deve fazer isso e não efetuar cálculos de valores que não estejam relacionados a isso.

Caso a instrução especificada em algum exercício deste capítulo já exista na linguagem em que estiver programando, é claro que você tem pelo menos duas opções: usar a disponível ou desenvolver a sua própria, para treinar. Neste caso, você pode comparar os resultados da sua com a da linguagem, para validar a instrução criada.

Nas especificações que se seguem, se for pedido "declare uma instrução" ou semelhante, você deverá fazer um pequeno programa para testá-la.

5.1.MATEMÁTICA

5.1.1 Número par - Fazer um procedimento que retorna Verdadeiro ou Falso conforme um número seja par ou não. Se necessário, convencione 0 e 1, "S" e "N" ou outra representação de Falso e Verdadeiro.

5.1.2 Numeração de 1 a 100 - Elaborar um procedimento que mostra os números de 1 a 100.

5.1.3 Numeraçãode N1 a N2 - Declarar um procedimento semelhante ao acima, mas que recebe como parâmetros os valores inicial e final.

5.1.4 Números pares - Escrever um procedimento que recebe dois números e mostra na tela os números pares situados entre os dois, inclusive. Testar com um programa que lê os números inicial e final e, se este maior que o inicial, chama o procedimento.

5.1.5 Equação do segundo grau - Implementar uma instrução que recebe os coeficientes a, b e c de uma equação do segundo grau e retorna suas raízes. Resolva: como fazer quando a equação resultante não tiver raízes reais?

5.1.6 Aprovação - Escrever uma função que recebe uma nota de 0 a 10 e retorna verdadeiro ou falso (ou outros valores convencionados como tal) se o aluno foi aprovado, isto é, se tirou 7 ou mais.

5.1.7 Maior e menor com menu - Faça duas funções: uma que recebe dois números e retorna o maior e outra que recebe o mesmo mas retorna o menor. Implementar um programa com um menu de 4 opções: ler dois números, testar uma e outra função e terminar.

5.1.8 Maior de 2 - Escrever uma função que recebe dois números quaisquer e retorna o maior.

5.1.9 Maior de 3 - Escrever uma função que recebe 3 números e retorna o maior

5.1.10 Entre 0 e 100 - Implementar uma função que recebe um número qualquer e retorna Verdadeiro se o número está entre 0 e 100, caso contrário retorna Falso.

5.1.11 Média de 3 - Declarar uma função que calcula a média aritmética de 3 números

5.1.12 Exponenciação inteira - Escrever uma função que calcula um inteiro elevado a outro inteiro, usando multiplicação.

5.1.13 Exponenciação real - Escrever uma função que calcula um número real elevado a outro real, usando multiplicação. Se houver tal função na linguagem em que estiver programando, compare seus resultados para vários tipos de valores.

5.1.14 Juros compostos - Sendo C o capital, n o prazo e i a taxa de juros, o valor futuro FV de uma aplicação financeira é calculado por:

$$FV = C(1+i)^n$$

Elabore um programa com duas opções: uma que, dados capital, prazo e taxa, calcule o valor futuro, e outra que, dados prazo, taxa e valor futuro, calcule o capital necessário.

5.1.15 Série - Elaborar programa que calcule, usando função, o valor da série abaixo para N termos, sendo N lido:

$$S = 1 - \frac{1}{3^2} + \frac{1}{5^2} - \frac{1}{7^2} + \frac{1}{9^2} - \Lambda$$

5.1.16 Bissexto - Um ano é bissexto se for divisível por 4 exceto os séculos, que são bissextos se forem múltiplos de 400. Implementar uma função que recebe o número de um ano e retorna Verdadeiro se o ano for bissexto ou Falso caso contrário.

5.1.17 Fatorial - Faça uma função que recebe como parâmetro um inteiro e retorna seu fatorial. O que você acha que poderia fazer para o caso em que o fatorial, se calculado, vai estourar a capacidade do tipo de dado adotado?

5.1.18 Números primos - Um número é dito ser primo quando é divisível somente por si e pela unidade. Faça um programa que verifica, através de uma função, se um número é ou não primo. *[Dica: divida o número N por todos os números de 2 a N - 1. Se o resto da divisão de N por algum dos números der zero, ele não é primo]*

5.1.19 Arco-tangente - O valor do arco-tangente pode ser calculado através da fórmula abaixo, válida quando $x^2 \pi 1$:

$$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \ldots$$

Elabore um programa que lê o valor de x e a quantidade de fatores, e compara o valor encontrado com o calculado pela respectiva instrução da linguagem. Teste para quantidades variadas de fatores.

5.2.CARACTERES E CADEIAS

5.2.1 Menor cadeia - Escrever uma função que recebe duas cadeias de caracteres e retorna a menor em ordem alfabética.

5.2.2 Leitura de cadeia - Implementar um procedimento que recebe uma linha e uma coluna e lê uma variável caractere na respectiva posição da tela, retornando o valor lido através de um parâmetro por referência ou equivalente.

5.2.3 Cabeçalho - Criar procedimento que recebe e imprime uma linha de cabeçalho com número de página, no formato. A data é obtida do sistema e o número de página é um parâmetro:

RELATÓRIO DE CLIENTES EMISSÃO: DD/MM/AAAA PÁG. 999

5.2.4 Dia, mês e ano - Implemente funções que recebem uma data no formato 'DD/MM/AAAA' e retornam dia, mês e ano, respectivamente.

5.2.5 Validação de data - Escrever uma função que verifica se uma data no formato 'DD/MM/AAAA' é válida. Além de verificar se o dia está entre 1 e 31 e o mês está entre 1 e 12, se o dia for 29/02 é chamada a função especificada no item 5.1.16 para verificar se o ano é bissexto.

5.2.6 Formatação de data - Elaborar um procedimento que recebe como parâmetros dia, mês, ano, uma linha e uma coluna da tela e uma letra que corresponde ao formato de data a ser apresentado. Se o formato = 'A', mostra DD/MM/AA; se 'B', mostra DD.MM.AAAA. Para testá-lo, faça um programa que busca a data do sistema, extrai dia, mês e ano e chama o procedimento.

5.2.7 Inversão de cadeia - Escrever uma função que recebe uma cadeia de caracteres e retorna-a invertida (lida de trás para a frente).

5.2.8 Palíndromos - Implementar um programa que verifica se uma frase é palíndroma (a mesma se lida normalmente ou de trás para a frente: "roma me tem amor", "socorram me subi no onibus em marrocos"). Use a função especificada acima e uma outra para retirar espaços de uma cadeia.

5.2.9 Ocorrências de subcadeia - Elabore uma instrução que identifica quantas vezes uma subcadeia ocorre em uma cadeia de caracteres (por exemplo, "na" ocorre duas vezes em "banana").

5.2.10 Maiúsculas - Implementar uma função que converte uma cadeia de caracteres para maiúsculas.

5.2.11 Minúsculas - Elaborar uma instrução que converte uma cadeia de caracteres para minúsculas. *[Dica: subtraia 32 dos caracteres cujos códigos ASCII estão entre 65 e 90, inclusive]*.

5.2.12 Inserção de caractere - Elaborar uma função que insere um caractere recebido entre cada letra de uma cadeia.

5.2.13 Trim - Implementar uma função que retira os brancos finais de uma cadeia de caracteres (em certas linguagens disponível com o nome de "Trim")

5.2.14 Ajuste de tamanho - Implementar uma função que insere brancos no fim de uma cadeia até atingir um tamanho especificado.

5.2.15 Crítica de data - Escrever uma função que recebe uma data no formato 'DD/MM/AAAA' (dia/mês/ano), verifica se as barras estão na posição correta, se o dia está entre 1 e 31 (inclusive), se o mês está entre 1 e 12 e retorna um valor da seguinte forma:

0: data correta
1: dia inválido
2: mês inválido
3: dia e mês inválidos
4: formato inválido

5.2.16 Formatação de linha - Um programador está escrevendo um editor de textos, e precisa de uma função que recebe uma linha de texto e uma largura de linha, e insere espaços de forma a alinhar o texto à largura recebida, retornando a linha formatada. Implemente essa função. *[Dica:*

primeiro calcule a quantidade de espaços necessária; descubra a quantidade de intervalos entre palavras e calcule quantos espaços terá que inserir em cada intervalo; insira essa quantidade em cada intervalo, sendo que o último intervalo receberá os espaços extras restantes. Investigue outras possibilidades de distribuição]

5.3.CONTROLE DE TELA

5.3.1 Linha vertical 1 - Desenvolva um procedimento que recebe um número de coluna de tela e preenche a coluna com caracteres 'O'.

5.3.2 Linha vertical 2 - Alterar o procedimento acima para que receba também o caractere.

5.3.3 Texto posicionado 1 - Escreva um procedimento que recebe duas coordenadas da tela (linha e coluna) e um texto, e mostra o texto na posição indicada.

5.3.4 Texto posicionado 2 - Altere o procedimento acima para que preserve a posição do cursor, salvando a linha e a coluna no início e restaurando-as no final.

5.3.5 Mensagem 1 - Escrever um procedimento que recebe valores de linha e coluna da tela e um texto, mostra o texto nas coordenadas recebidas e espera 5 segundos. Após, apaga a mensagem e termina.

5.3.6 Mensagem temporizada - Alterar o procedimento acima para receber também o tempo a esperar e, se for pressionada uma tecla, terminar.

5.3.7 Entrada de cadeias - Elabore uma instrução que efetua leituras de cadeias de caracteres. Ela recebe linha e coluna, o "prompt" (texto que aparece antes, como 'Nome: ' ou 'CPF: '), o tamanho máximo da cadeia e o valor inicial. A instrução controla o uso das setas à esquerda e à direita, permite Delete e Backspace, Home e End. Ela termina com Enter ou Escape, neste segundo caso restaurando o valor inicial.

5.3.8 Entrada de números - Desenvolva uma versão da instrução especificada acima que permite uma opção de só aceitar dígitos.

5.4.SONS

5.4.1 Som crescente - Elaborar um programa que emite sons de freqüência crescente, de uma freqüência inicial até uma final, com percentual de variação e duração de cada som lidos pelo teclado. Para emitir o som use um procedimento que recebe esses dados como parâmetros. Critique o percentual de variação, impedindo valores (como 0) que atrapalhem a execução.

5.4.2 Freqüência de notas musicais - Se você conhece a freqüência de uma nota musical, pode conhecer a nota seguinte (um semitom acima) multiplicando aquela por $\sqrt[12]{2}$. Sabendo que a freqüência de uma das notas Lá é 440 Hz, faça o seguinte:

Escreva uma função que recebe uma nota e retorna a próxima;

Escreva um programa que calcula as freqüências de notas acima do Lá (La#, Si, Dó, Dó#, Ré, Ré#, Mi, Fá, Sol, Sol#, Lá, Lá #, Si, Dó. Anote as freqüências.

Em outro programa, declare constantes para as freqüências das notas (por ex., DO) e escreva instruções que toquem o Parabéns prá você. *[Dica: declare uma constante para a duração e use múltiplos desta; as primeiras notas, com a respectiva duração, do Parabéns ficariam assim: (DO, Dur), (DO, Dur), (RE, 4*Dur), (DO, 2*Dur). Tente Dur = 100 e 200.]*

Comentario: *Obs.: frequência do primeiro Lá: 27,5. Próximos Lá = 27,5 * 2**N, onde N é a oitava desejada (pag. 207 do Scheid).*

5.5.REGISTROS E VETORES

5.5.1 Nome do mês - Escrever uma função que recebe número do mês e retorna seu nome por extenso.

5.5.2 Nome do dia - Idem acima, para o dia da semana.

5.5.3 Maior e menor - Escrever um programa com quatro opções (implementadas através de instruções declaradas): preencher um vetor de 10 elementos com valores inteiros aleatórios, mostrar o conteúdo do vetor, identificar o maior e o menor número gerados, e respectivas posições.

5.5.4 Crítica de dia do mês - Alterar a função acima para que verifique se o dia está compatível com o mês. Por exemplo, novembro não possui dia 31. Para isso use também uma função.

5.5.5 Estatística de notas - Escrever um programa para fazer estatísticas de notas de um aluno. O programa oferece opções de: entrar com nomes de disciplinas e respectivas notas, achar a maior nota, achar a menor nota e calcular a média das notas. Quando mostra algum resultado, o programa espera ser teclado algo para oferecer novamente o menu. Sugestões para modularização:

- Mostrar o menu, ler e retornar a opção (função)
- Ler os valores das notas, disciplinas e retornar a quantidade (procedimento)
- Identificar a posição do menor (função)
- Identificar a posição do maior (função)
- Calcular a média (função)

5.5.6 Estatística de notas 2 - Alterar o programa acima para impedir qualquer cálculo se os vetores estiverem vazios.

5.5.7 Estatística de notas 3 - No mesmo programa acima, incluir opções para

a) Mostrar o conteúdo atual dos vetores de notas e disciplinas
b) Dada uma disciplina, pesquisar a nota correspondente
c) Dada uma nota, pesquisar se ela existe no vetor de notas e mostrar a respectiva disciplina

5.5.8 Quadrado mágico - Um *quadrado mágico* é aquele dividido em linhas e colunas, com um número em cada posição e no qual a soma das linhas, colunas e diagonais é a mesma. Por exemplo, veja um quadrado mágico de lado 3, com números de 1 a 9:

8	3	4
1	5	9
6	7	2

Elabore um programa que identifica e mostra na tela todos os quadrados mágicos com as características acima. Analise os quadrados identificados e verifique se há alguma diferença básica entre eles ou se podem ser considerados os mesmo sob algum aspecto. *[Dica: produza todas as combinações possíveis e verifique a soma quando completar cada quadrado. Usar um vetor de 1 a 9 (a estrutura que usei) parece ser mais simples que usar uma matriz 3x3]*

5.6.ARQUIVOS

5.6.1 Salvamento de parâmetros - Implementar um módulo de gravação de parâmetros com duas possibilidades: salvar ou recuperar o valor de um parâmetro. Um parâmetro é identificado através de um nome de até 8 caracteres. O valor pode ser uma cadeia de até 255 caracteres.

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

5.6.4 Configuração de impressão - Elaborar uma instrução que grava em disco um registro com informações sobre configuração de impressão: nome da impressora, largura e altura do papel, margens (superior, inferior, direita, esquerda). Escrever outra instrução que recupera os dados gravados.

5.6.5 Banco de palavras - Montar um módulo com instruções para manter um banco de palavras. Deve haver instruções para incluir, alterar e excluir uma palavra do banco, além de pesquisar se uma dada palavra está no banco. O banco é armazenado em um arquivo em disco. Todas as instruções devem ser reutilizáveis. Para testar, faça um programa simples com uma opção para cada operação que pode ser feita no banco.

5.6.6 Contador - Declarar instruções para manter contadores, cujos valores atuais são armazenados em disco. Isto serve, por exemplo, para designar códigos de clientes ou produtos no caso de numeração sequencial. Prever instruções para criar um contador (zerado) e recuperar o próximo número (que também atualiza o valor atual). Para identificar cada contador use o nome do arquivo.

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

5.6.8 Arquivo de senhas - Uma pessoa pode ter inúmeras senhas: da conta corrente, poupança, da conta do outro banco, cartão de crédito, provedor Internet, da rede local do trabalho e por aí vai. Faça uma programa que permita gravar várias senhas, com uma descrição de cada uma. Inclua a possibilidade de mostrar na tela a lista das senhas.

5.7.VARIADOS

5.7.1 Maior qualquer - Implementar uma função que recebe dois valores de qualquer tipo de dado (cadeia, número inteiro ou real, caractere) e retorna o maior. *[Verifique se a linguagem permite parâmetros sem tipo]*

5.7.2 Dia da semana - Descubra como, a partir de uma data válida, você pode identificar o dia da semana correspondente (domingo, segunda, etc.). Escreva uma função que retorna esse dia.

5.7.3 Crítica completa de data - Combinar as especificações 5.2.15 e 5.5.4 para formar uma crítica de data mais completa.

5.7.4 Custo de execução de procedimento - Escrever um programa que serve para se medir o custo, em tempo, de execução de um procedimento. Ele troca, alguns milhares de vezes, os valores de duas variáveis, de duas formas: na primeira é usado para trocar as variáveis um procedimento, na segunda sem este. Computar o tempo gasto para cada forma e mostrá-los na tela.

5.7.5 Sorteio de dados - Escreva uma instrução que recebe um número de 1 a 6, correspondente ao sorteio de um dado, e desenha o dado na tela (em qualquer posição), mostrando o lado sorteado. Depois, faça um programa que sorteia 5 dados e os mostra na tela, alinhados.

5.7.6 PIS/PASEP - Escrever uma função que recebe um número de PIS/PASEP e retorna o dígito verificador (veja a regra na especificação 3.6.3).

5.7.7 CPF - Escrever função para calcular os dígitos de controle do CPF (regra na especificação 4.8.8). Para simplificar, já que são duas somatórias, escreva também uma função auxiliar que recebe o CPF e o peso inicial e retorna a soma.

5.7.8 Palavra grande - Elabore uma instrução que desenha "grande" uma letra do alfabeto, em linha e coluna da tela. Outra instrução recebe um texto de até 10 caracteres e chama a primeira para mostrar o texto na tela em letras grandes.

5.7.9 Reprodução de melodia - Declarar uma instrução que recebe um vetor de pares freqüência/duração, e reproduz os sons na seqüência do vetor. Valor zero para a freqüência representa uma pausa.

5.7.10 Melodias em arquivos - Elaborar um programa que toca músicas, com opções para: editar uma melodia (nova ou existente) e salvá-la em disco, reproduzir ou eliminar uma melodia gravada. Cada melodia é armazenada em um vetor com freqüência e duração de cada nota ou pausa (freqüência 0).

5.7.11 Implementar uma instrução que recebe uma cadeia e retorna-a embaralhada. Usar números aleatórios.

5.7.12 Janela de confirmação - Implementar uma instrução que mostra uma janela de confirmação na tela com as opções "SIM", "NÃO" e "CANCELAR", aguarda a opção do usuário e retorna a opção selecionada. O usuário seleciona a opção com as setas ou a primeira letra e depois teclando Enter.

5.7.13 Relatório de notas - Implementar um programa com opções para cadastrar (incluir ou acrescentar) um arquivo contendo nomes e notas de alunos e para imprimir um relatório dos dados gravados. O relatório deve ter um cabeçalho contendo data e hora de emissão, nome do relatório, numeração de páginas e quebra (mudança) a cada 66 linhas. Ao final é mostrada a quantidade e a média das notas.

6.IDÉIAS E MAIS IDÉIAS

Neste capítulo você terá várias idéias para trabalhos práticos. Constituem especificações mais complexas que as do restante do texto, e que certamente o desafiarão. A maioria delas exigirá algum detalhamento extra, isto é, a especificação não é completa e você deverá suprir detalhes. Particularmente procure aplicar conceitos de reutilização e modularização.

Alerta: as especificações da seção Desafios não são para iniciantes!

6.1.ENTRETENIMENTO

6.1.1 Forca - Implementar um programa que jogue o jogo da forca. 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 perdidas, 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.

6.1.3 Jogo-da-velha - Elaborar um programa que jogue o jogo-da-velha, com opções de controlar dois jogadores ou jogar o computador contra um jogador.

6.1.4 Combinações de letras - Implementar um programa que lê uma palavra de 4 letras e gera todas as combinações possíveis das quatro letras, sem repetição. O programa deve fornecer um menu para o usuário, permitindo:

- entrar nova palavra
- gerar combinações
- mostrar na tela (formatadas em colunas)
- imprimir (também em colunas)
- eliminação de palavras indesejadas (por exemplo, que não existam).

Extensões: gravar em arquivo as palavras encontradas (um arquivo para cada palavra-chave), e permitir ao usuário recuperá-las. Permitir qualquer quantidade de letras na palavra. Descobrir como é estruturado o arquivo de palavras de algum dicionário ou processador de textos, e usá-lo para filtrar palavras existentes.

6.1.5 Bingo - Elabore um programa que faz sorteios de bingo. O programa deverá oferecer opções de iniciar um sorteio, sortear um número e apresentar os números sorteados até um determinado momento. Note que o programa não poderá repetir um número já sorteado.

6.1.6 Arquivo de senhas - Uma pessoa pode ter inúmeras senhas: da conta corrente, poupança, da conta do outro banco, cartão de crédito, provedor Internet, da rede local do trabalho e por aí vai. Faça uma programa que permita gravar várias senhas, com uma descrição de cada uma. Inclua opções para mostrar na tela ou imprimir a lista das senhas e para pesquisar por uma palavra chave da descrição.

6.1.7 Desenho datilográfico - Implementar um programa que permite o uso de toda a tela para desenhos utilizando caracteres disponíveis no teclado. Inclua opções para salvar uma tela, carregar um desenho salvo e ajuda para os comandos.

6.1.8 Ping-Pong - Talvez você não conheça, mas um dos primeiros videogames era um Philco monocromático, e um dos jogos, chamado de Ping-Pong, tinha duas "raquetes" que se moviam na vertical das laterais da tela e uma "bolinha", cuja velocidade aumentava depois de algumas "raquetadas" dos jogadores. Um jogador fazia um ponto quando o adversário deixava a bolinha passar; quem fizesse 15 pontos primeiro ganhava. Implemente esse jogo.

6.1.9 Meteoros: o jogo - Uma nave espacial, embaixo na tela, deve ultrapassar um campo de meteoros (por exemplo, se em modo texto, asteriscos), desviando-se à esquerda ou direita. Se tocar em algum, ela explode. Cada "linha" de meteoros ultrapassada conta um ponto, e a velocidade dos meteoros vai aumentando, digamos, a cada 500 pontos. Implementar o jogo. Entre outras coisas, o programa deverá preencher a próxima linha de meteoros, no alto da tela, e mover esta uma linha para baixo (supondo ainda que a tela estará em modo texto), além de verificar se houve colisão.

6.1.10 Monitor YAM - Talvez o jogo YAM ainda esteja disponível comercialmente, para você obter as regras e implementar um programa que controle o jogo para duas ou mais pessoas. O programa deverá sortear os dados, mostrar a tabela e criticar as escolhas de um jogador: se ele fez quadra, por exemplo, só pode inserir os pontos em um dos locais apropriados da tabela.

6.2.DOMÉSTICOS

6.2.1 Agenda telefônica - Implemente um programa que controla uma lista de nomes e telefones, com opções para incluir, alterar, excluir e pesquisar por nome ou por telefone.

6.2.2 Controle de filmes - Faça um programa que mantém dados a respeito dos filmes assistidos por uma pessoa: nome, diretor, roteirista(s), atores, data, comentários. Inclua consultas que julgar úteis.

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*.

6.2.4 Despensa - Implemente um controle simples de mercadorias em uma despensa doméstica. Sobre cada produto podem ser armazenados um código numérico, descrição e quantidade atual. O programa deve ter opções para entrada e retirada de produtos, bem como um relatório geral e um de produtos não disponíveis.

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.).

6.2.7 Etiquetas para cheques - Faça um programa que preenche mini-etiquetas para afixar em folhas de cheque, contendo nome, endereço, telefone e RG. Provavelmente você terá que pesquisar os comandos de movimentação do carro da impressora, para obter ajuste fino e assim manter o posicionamento correto.

6.2.8 Histórico do automóvel - Registre tudo que ocorrer com o carro: manutenções, combustível, lavagens, etc., tudo com o valor gasto e a quilometragem. Além de um relatório geral, inclua uma consulta sobre custo por quilômetro e consumo médio.

6.2.9 Manutenções domésticas - Se você cuida de uma casa ou apartamento, sabe que vez por outra são necessários alguns serviços de manutenção: pia que vaza, cadeira que quebra, cortinas que cedem, armários que desajustam etc., etc. Faça um programa que registre o tipo de serviço, data e nome de quem fez, prazo de garantia e observações, como por exemplo sobre as chamadas para consertar o conserto!

6.3.EDUCATIVOS

6.3.1 Tabuada - Implementar um programa para crianças que apoia o aprendizado de tabuada. Entre outras coisas, o programa pode propor à criança seqüências de um mesmo número ou aleatórias. Pode também mostrar números em tamanho maior. Se quiser sofisticar mesmo, armazene estatísticas de erros e acertos, e faça o programa reforçar os pontos fracos.

6.3.2 Código Morse - Implemente um tradutor de código Morse. Pode ter opções para som, imagens ou listagens. Dizem que só a Marinha ainda usa esse código, mas pode ser um bom exercício...

6.4.VARIADOS

6.4.1 Copa do Mundo - Fazer um programa para controle do placar de uma Copa do Mundo de futebol (escolha uma). O programa deve ler e armazenar os resultados, além de montar os jogos a partir das oitavas-de-final

6.4.2 Lista de seleção - Criar uma instrução reutilizável que recebe como parâmetro uma lista de strings, abre uma moldura centralizada na tela, mostra as strings e permite seleção de uma (ou ESCAPE). Quando selecionada alguma, a instrução retorna o número dela na lista ou zero, se nenhuma selecionada.

Para testá-la, implemente um programa simples, por exemplo, um que leia os valores a selecionar do teclado.

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.

6.4.4 Histórico escolar - Faça um programa que registra o histórico escolar de um ou mais estudantes. Guarde informações sobre período ou série, disciplinas e notas, e calcule média em período e geral. Se a sua escola não trabalha com notas e sim com menções ou algo parecido, calcule as quantidades de cada menção.

6.4.5 Controle de backup - Implemente um programa que controla o cadastro de disquetes: inclusão, alteração, exclusão, consultas. Para cada disquete, são armazenados: número (que deve corresponder a um número no disquete), data da compra, descrição do conteúdo. Você pode incluir também a data do último "refresh" (vamos chamar assim a regravação do disquete para garantir que o backup continua íntegro), com uma consulta para mostrar os disquetes vencidos. Como os disquetes tem normalmente garantia "eterna", será interessante armazenar o número da nota fiscal da compra, para o caso de troca se o dito cujo se tornar defeituoso (o que você já deve ter observado que acontece com razoável freqüência).

6.4.6 Avaliador de expressões aritméticas - Faça um programa que calcula qualquer expressão aritmética contendo as operações básicas (+, -, /, *), constantes numéricas reais e parênteses. Usar recursividade vai tornar as soluções bastante elegantes.

6.4.7 Problemas e soluções - No caminho do aprendizado temos que resolver ou descobrir solução para vários tipos de problemas, relacionados ao computador, sistema operacional, impressora, etc.

Faça um programa para registrar essa experiência. Ele armazena textos descritivos a respeito do problema e respectiva solução, permitindo pesquisa por palavra chave.

6.5.DESAFIOS

6.5.1 YAM inteligente - O monitor deste jogo (6.1.10) era mais fácil. Agora o desafio é implementar a possibilidade de um jogador jogar contra o computador.

6.5.2 Grade horária - Faça um programa que monte grades horárias a partir de horários, disciplinas, professores e restrições, solicitações e outras variáveis. Há coordenadores que aguardam ansiosamente tal programa...

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.

APÊNDICE A: DESC

A tabela abaixo constitui um sumário das instruções necessárias à implementação dos programas especif estrutura do texto. Uma das colunas indica o nome da instrução no Turbo Pascal 7.0; a coluna em branco pode ser u outra linguagem, com a mesma finalidade.

Note que este não é um guia de sintaxe, apenas uma referência para relacionar uma finalidade a uma instrução.

Obs: Instruções assinaladas com (f) são funções (retornam um resultado). As demais são procedimentos.

SEÇÃO	INSTRUÇÕES	TURBO PASCAL 7.0	
GERAL	Como digitar, salvar, compilar e executar o programa fonte Como executar passo-a-passo o programa e observar variáveis	Ambiente Integrado de Desenvolvimento	
SAÍDA SIMPLES	Mostrar na tela Mostrar na impressora	Write, Writeln Write(LST,...)	
CONTROLE DE TECLADO E TELA (modo texto)	Limpar a tela Posicionar cursor Obter coluna do cursor Obter linha do cursor Alterar cor de fundo Alterar cor do texto Verificar se uma tecla foi pressionada Obter o caractere correspondente à tecla pressionada Definir uma janela de texto na tela	ClrScr GotoXY, CursorTo WhereX (f) WhereY (f) TextBackGround TextColor KeyPressed (f) ReadKey Windows	
SONS	Emitir som Pausa Interromper som	Sound Delay NoSound	

SEÇÃO	INSTRUÇÕES	TURBO PASCAL 7.0	
MEMÓRIA E VARIÁVEIS	declarar variáveis	VAR	
	declarar constantes	CONST	
	como atribuir valores a variáveis	NomeDaVariável := expressã	
	como recuperar valores de variáveis	Nome da variável	
	como ler valores do teclado e armazená-los em variáveis	Readln	
MATEMÁTICA	Aritmética básica	+, -, *, /	
	Resto, quociente inteiro	DIV, MOD	
	funções matemáticas (seno, cosseno, exponencial, raiz quadrada, elevar ao quadrado, truncar, arredondar, valor absoluto, parte fracionária, parte inteira, logaritmo neperiano)	Sin, Cos, Exp, SqRt, Sqr, Trı Round, Abs, Frac, Int, Ln	
	Incrementar, decrementar uma variável	Inc, Dec	
CARACTERES E CADEIAS	Concatenação de cadeias	+, Concat	
	Obter um caractere de uma cadeia	NomeDaVariável[Posição]	
	Obter o tamanho atual de uma cadeia	Length (f)	
	Código ASCII de um caractere	Ord (f)	
	Caractere referente a um código ASCII	Chr (f), #	
	Pesquisar uma cadeia em outra	Pos (f)	
	Converter cadeia com dígitos em número	Val	
	Converter número em cadeia de caracteres	Str	
	Converter letra para maiúscula	UpCase (f)	
	Inserir uma cadeia em outra	Insert	
	Extrair uma subcadeia de uma cadeia	Copy (f)	
	Excluir uma subcadeia de uma cadeia	Delete	
	Preencher uma cadeia com uma quantidade de caracteres	FillChar	
ALTERNATIVAS E DECISÃO	Executar condicionalmente trechos de instruções	IF..THEN..ELSE CASE..DO	

SEÇÃO	INSTRUÇÕES	TURBO PASCAL 7.0	
REPETIÇÃO	Pelo menos uma vez, quantidade indefinida de repetições	REPEAT..UNTIL	
	Quantidade conhecida de repetições	FOR..TO..DO	
		FOR..DOWNTO..DO	
	Qualquer situação	WHILE..DO	
VETORES E MATRIZES	declarar vetores ou matrizes	ARRAY	
	referenciar elemento de vetor	NomeVar[Pos1, Pos2...]	
ARQUIVOS	declarar variáveis de arquivos	tipos de dado File, Text	
	associar nomes de variáveis a nomes de arquivos em disco	Assign	
	criar arquivos	Rewrite	
	abrir arquivos para leitura e/ou escrita	Rese	
	gravar dados em arquivos	Write, BlockWrite	
	Ler dados de arquivos	Read, BlockRead	
	Fechar um arquivo	Close	
	Obter status da última operação com arquivo	IOResult	
	Eliminar um arquivo	Erase	
	Posicionar o apontador de registros	Seek	
	Obter a posicão atual do apontador de registros	FilePos (f)	
	Obter a quantidade de registros de um arquivo	FileSize (f)	
	Verificar se o apontador de registros atingiu o fim do arquivo	EOF (f)	
	Renomear um arquivo	Rename	
	Obter o diretório corrente	GetDir	
	Alterar o diretório corrente	ChDir	
	Criar um subdiretório	MkDir	
	Eliminar um subdiretório	RmDir	
	Truncar um arquivo	Truncate	

SEÇÃO	INSTRUÇÕES	TURBO PASCAL 7.0	
DECLARAÇÃO DE INSTRUÇÕES	criar (declarar) procedimentos	PROCEDURE	
	criar (declarar) funções	FUNCTION	
	chamar instruções declaradas	nome da instrução	
	como declarar parâmetros de entrada em instruções declaradas	no cabeçalho, nome e tipo de	
	como declarar parâmetros de entrada e saída em instruções declaradas	antepor a palavra var	
	chamar (executar) uma instrução declarada, passando parâmetros para as instruções que criou e recebendo valores de retorno.	nome da instrução, com parâ entre parênteses	
OUTRAS	Suspender o fluxo de execução por um tempo determinado	Delay	
	Interromper o programa	Halt	
	Gerar um número aleatório	Randomize, Random (f)	
	Obter data e hora	GetDate, GetTime	

Comentario: **BIBLIOGRAFIA**

CARVALHO, Ricardo F. *Borland Turbo Pascal 6.0*. Berkeley, 1992

FARRER, Harry et alli. *Pascal Estruturado*. Guanabara Koogan, 1995

*FORBELLONE, André Luiz V. & EBERSPÄCHER, Henri F. *Lógica de Programação*. Makron, 1993.

*GUIMARÃES, Ângelo M. & LAGES, Newton A. C. *Algoritmos e Estruturas de Dados*. LTC, 1985.

HERGERT, Douglas. *Dominando o Turbo Pascal 5*. Ciência Moderna, 1989.

KERNIGHAM, W. *Ferramentas para a programação em Pascal*. Campus - 1988.

MECLER, Ian & Maia, Luiz Paulo. *Programação e Lógica com Turbo Pascal*. Campus, 1989.

*O'BRIEN, Stephen. *Turbo Pascal 6.0 Completo e Total*. Makron

SCHEID, Francis. *Computadores e Programação*. McGraw-Hill, 1984.

SCHILDT, Herbert. *Turbo Pascal Avançado - Guia do Usuário (até 6.0)*. Makron, 1988

SCHMITZ, Eber Assis & TELES, Antonio A. S. *Pascal e Técnicas de Programação*. LTC, 1986.

SYCK, Gary. *Turbo Pascal Soluções*. Campus.

*WEISKAMP, Keith. *Turbo Pascal 6.0*. LTC, 1992.

WIRTH, Niklaus. *Programação Sistemática em Pascal*. Campus, 1987.