

Sequenciais

S000000050 - Faça um algoritmo que leia um valor N, representando o lado de um quadrado, e calcule e escreva a área do quadrado.

S000000100 - Faça um algoritmo que leia dois valores inteiros e escreva a sua soma.

S000000150 - Faça um algoritmo que leia 3 valores reais, notas de um aluno, e escreva sua média aritmética. A média aritmética de um conjunto de valores é dada pela soma dos valores dividido pela quantidade de valores considerados.

S000000200 - Faça um algoritmo que leia 2 valores reais v1 e v2 e calcule e escreva a área do triângulo que tem base igual a v1 e altura igual a v2. Dica: A área de um triângulo é dada pela expressão: $(\text{base} \times \text{altura})/2$

S000000210 - O produto vetorial $A \times B$ de dois vetores $A=[ax, ay]$ e $B=[bx, by]$ é um vetor perpendicular ao plano formado pelos vetores A e B, de módulo igual a $|ax \cdot by - ay \cdot bx|$. Faça um algoritmo que leia as componentes ax, ay, bx e by dos vetores A e B, e calcule e escreva o módulo do vetor resultante do produto vetorial.

S000000220 - O produto escalar $A \cdot B$ de dois vetores $A=[a1, a2, a3...]$ e $B=[b1, b2, b3...]$ é dado pelo somatório $a1 \cdot b1 + a2 \cdot b2 + a3 \cdot b3 + ...$. Faça um algoritmo que leia as coordenadas ax, ay, az, bx, by, bz de dois vetores no espaço tridimensional e calcule e escreva seu produto escalar.

S000000300 - Faça um algoritmo que leia 3 valores a, b e c, coeficientes de uma equação de segundo grau, e calcule e escreva a soma das raízes da equação. Dica: As raízes de uma equação podem ser calculadas pela fórmula de Baskhara.

S000000350 - Faça um algoritmo que leia 3 valores a, b e c, coeficientes de uma equação de segundo grau, e calcule e escreva o produto (multiplicação) das raízes da equação. Dica: As raízes de uma equação podem ser calculadas pela fórmula de Baskhara.

S000000400 - Faça um algoritmo que leia 2 valores e escreva o maior deles. Dica: O maior entre dois valores quaisquer v1 e v2 pode ser calculado pela expressão $(v1 + v2 + |v1 - v2|)/2$. O operador matemático de módulo ($|x|$) é representado na nossa linguagem pela função `abs(x)`

S000000500 - Faça um algoritmo que leia 2 valores e escreva o menor deles. Dica: O menor entre dois valores pode ser obtido a partir do maior valor e da soma dos dois (por exemplo: Se a soma de dois valores é 15 e o valor do maior é 10, como pode ser calculado o valor do menor?).

S000000600 - Faça um algoritmo que leia 3 valores escreva o maior deles. Dica: O maior entre três valores pode ser encontrado pela aplicação repetida da expressão para encontrar o maior entre dois valores.

S000000700 - Faça um algoritmo que lê um valor inteiro em reais e calcula e escreve qual o menor número possível de notas de 100, 50, 20, 10, 5, 2 e 1 real em que o valor pode ser decomposto. Dica: Isso pode ser calculado a partir de operações de divisão inteira.

S000000800 - Faça um algoritmo que lê uma quantia inteira em segundos e escreva o número de horas, minutos e segundos correspondente. Dica: Para converter segundos para minutos usa-se a divisão inteira por 60.

S000000900 - Faça um algoritmo que lê 3 valores, lados de um triângulo, e calcule e escreva a área do triângulo formado. Dica: A área de um triângulo de lados l1, l2 e l3 pode ser calculada pela expressão $\text{Área} = \text{raiz}(S \cdot (S - l1) \cdot (S - l2) \cdot (S - l3))$, onde S é o semi-perímetro, ou seja, a metade da soma dos lados

S00001000 - Faça um algoritmo que le um valor entre 0 e 9999 e calcula a soma dos seus dígitos. Dica: O dígito menos significativo de um número inteiro pode ser obtido pelo resto da divisão do número por 10. Os dígitos restantes podem ser obtidos pela divisão inteira por 10.

S00001100 - Faça um algoritmo que leia 3 valores v1, v2 e v3, e troque os valores entre si de modo que ao final a variável v2 contenha o valor que foi lido para v1, a variável v3 contenha o valor que foi lido para v2, e a variável v1 contenha o valor que foi lido para a variável v3. Dica: Para trocar os valores de duas variáveis (digamos v1 e v2) entre si, salva-se o valor de uma delas (p.ex., v1) em uma variável auxiliar qualquer (p.ex., aux), copia-se o valor da segunda (v2) para a variável que foi salva (v1), e copia-se o valor da variável auxiliar (que contém o valor original da primeira), para a segunda variável.

S00001200 - Faça um algoritmo que leia 4 valores, Hi, Mi, Hf, Mf, representando respectivamente a hora e minuto inicial e final de um evento, e calcule a duração do mesmo em horas e minutos. Considere que o evento inicia e termina no mesmo dia. Dica: Para simplificar o problema, converta cada par de valores em um único valor em minutos.

S00001300 - Faça um algoritmo que leia dois horários (hora, minuto e segundo) e escreva quantos segundos transcorreram entre esses dois horários.

S00001350 - Faça um algoritmo que leia um horário (hora, minuto e segundo) e escreva o horário correspondente ao segundo seguinte. Ex: se for lido, 17,21,36 escrever 17,21,37. Considere que os horários (lido e escrito) estão na faixa de 00:00:00 a 23:59:59.

S00001400 - Faça um algoritmo que a partir de um horário (hora, minuto, segundo) e uma quantidade de segundos transcorridos, calcule o segundo horário.

S00001500 - Faça um algoritmo que leia a quantidade de alunos em uma sala de aula e a quantidade de alunos por grupo, e calcule e escreva quantos grupos serão formados e o resto de alunos que não foram suficientes para formar mais um grupo.

S00001600 - Faça um algoritmo que leia um valor em binário até 6 dígitos e escreva o número correspondente em decimal.
Dica: Multiplique cada dígito binário pelo seu peso (1,2,4,8...) e some os produtos.

S00001650 - Faça um algoritmo que leia um número de 10 dígitos e rotacione os dígitos uma posição para a esquerda, de modo que o primeiro dígito passe a ocupar a última posição. Ex: 1234561234 deve gerar 2345612341 (todos os dígitos foram uma posição para a esquerda e o 1 da primeira posição passou a ocupar a última posição).

S00001700 - Faça um algoritmo que leia 5 números inteiros e escreva a quantidade deles que são pares. Dica: Um número é par se o resto da sua divisão por 2 é zero.

S00001800 - Faça um algoritmo que leia 5 números inteiros e escreva a soma dos que forem ímpares. Dica: Um número é ímpar se o resto da sua divisão por 2 é um.

Condicionais

C00000020 - Faça um algoritmo que leia um valor e escreva: 0, se o valor é zero; 1, se o valor é maior que zero; -1 - se o valor é negativo.

C00000040 - Faça um algoritmo que leia um valor e escreva: 0, se o valor é par; 1, se o valor é ímpar.

C00000060 - Faça um algoritmo que leia dois valores e, através de uma comparação, escreva o maior deles. Considere que os dois valores são diferentes.

C000000080 - Faça um algoritmo que leia dois valores e escreva os dois em ordem crescente.

C000000100 - Faça um algoritmo que leia 3 valores v1, v2 e v3, e escreva-os em ordem crescente.

C000000200 - Faça um algoritmo que leia 3 valores v1, v2 e v3 e coloque-os em ordem crescente, de forma que v1 contenha o menor, v2 contenha o elemento do meio (nem o maior, nem o menor), e v3 contenha o maior. Escreva os valores ordenados.

C000000250 - Escreva um algoritmo que leia os valores das quatro provas de um aluno e escreva a média aritmética considerando apenas as três melhores notas. Por exemplo, se os valores lidos foram 9, 9.5, 7, e 8, a média será $(9 + 9.5 + 8)/3$ (a prova de nota 7 é descartada). Dica: Não esqueça de considerar a possibilidade de ocorrerem notas iguais.

C000000300 - Faça um algoritmo que leia 3 valores a, b e c, coeficientes de uma equação de segundo grau, e verifique se a equação tem raízes reais. Se a equação tiver raízes reais, calcule e escreva as raízes da equação (em ordem crescente). Se não tiver, escreva "A equação não possui raízes reais". Dica: As raízes de uma equação podem ser calculadas pela fórmula de Baskhara. Uma equação não possui raízes se reais se $B^2 - 4*a*c < 0$

C000000350 - Faça um algoritmo que leia 3 valores a, b e c, lados de um triângulo, e verifique o tipo de triângulo formado escrevendo: 0 - se o triângulo é equilátero (os três lados são iguais); 1 - se o triângulo é isósceles (dois lados iguais e um diferente); 2 - escaleno (3 lados diferentes).

C000000360 - Faça um algoritmo que leia 3 valores a, b e c, lados de um triângulo, e verifique o tipo de triângulo formado escrevendo: 0 - se o triângulo é retângulo ($A^2 = B^2 + C^2$); 1 - se o triângulo é acutângulo ($A^2 > B^2 + C^2$); 2 - obtusângulo ($A^2 < B^2 + C^2$). Lembre que, para aplicar as relações mostradas, o algoritmo deve garantir que o maior dos 3 lados estará em A.

C000000400 - Faça um algoritmo que leia 3 valores l1, l2 e l3 e verifique se formam um triângulo. Se formarem, calcule e escreva a área do triângulo formado (veja exercício S000000900). Se não formarem, escreva -1.

C000000410 - Faça um algoritmo que leia 3 valores l1, l2 e l3 e verifique se formam um triângulo e, se formarem, o tipo de triângulo formado, escreva: 0 - se não formarem triângulo; 1 - se formarem um triângulo equilátero (os três lados são iguais); 2 - se formarem um triângulo isósceles (dois lados iguais e um diferente); 3 - se formarem um triângulo escaleno (3 lados diferentes)

C000000450 - Faça um algoritmo que implemente uma calculadora de 4 operações. O algoritmo deve ler, nessa ordem: o primeiro operando, o operador (+, -, *, /) e o segundo operando, e deve escrever o resultado da operação. Por exemplo, se o usuário digitar 2, +, 3 o algoritmo deve escrever 5. Dica: Para que uma variável possa receber um texto (mesmo que seja apenas o caracter "+") ela deve ser declarada como "literal".

C000000460 - Faça um algoritmo que leia um literal contendo uma expressão com 3 caracteres, onde o primeiro caractere é um dígito entre 0 e 9, o segundo é um operador (+, -, *, /) e o terceiro caractere é um dígito entre 0 e 9, e efetue o cálculo da expressão escrevendo o resultado. Dica: Use a função subliteral para separar os 3 caracteres e a função val para converter os operandos em seus valores numéricos.

C000000500 - Faça um algoritmo que leia 3 notas de um aluno e escreva sua média harmônica. Dica: A média harmônica entre três valores N1, N2 e N3 é calculada pela expressão $3/(1/N1 + 1/N2 + 1/N3)$. O que acontece se alguma das notas for igual a 0? Que resultado o algoritmo deve emitir?

C00000600 - Faça um algoritmo que leia 3 notas de um aluno e escreva sua média harmônica. Se o aluno obteve média abaixo de 6.0, E SOMENTE NESSE CASO, leia uma quarta nota (da prova de recuperação) e substitua a menor das três notas pela nota da recuperação e recalcule a média harmônica. Escreva a média harmônica final e o conceito obtido (0, se média harmônica (MH) < 6.0; 1 se 6.0 <= MH < 7.0; 2 se 7.0 <= MH < 8.0; 3 se 8.0 <= MH < 9.0; 4 se MH >= 9.0).

C00000625 - Faça um algoritmo que leia os códigos e as votações dos 5 candidatos da eleição para presidente em 1o turno. O algoritmo deve escrever, para cada candidato, seu código e seu percentual de votos. Se a eleição foi decidida no 1o turno, i.e., um candidato obteve 50% dos votos válidos + 1, o algoritmo deve escrever o código e o percentual de votos do eleito. Em caso contrário, o algoritmo deve escrever os códigos e os percentuais de votos dos dois candidatos que disputarão o segundo turno.

C00000650 - Faça um algoritmo que leia 3 notas de um aluno e escreva sua média harmônica (MH). Caso alguma das notas seja 0 (zero), MH deve receber 0 (zero). Se o aluno obteve uma MH abaixo de 6.0 e AINDA PUDER OBTER média igual ou superior a 6.0, o algoritmo deve ler uma quarta nota (da prova de recuperação) e substituir a menor das três notas pela nota da recuperação, recalculando MH. O algoritmo deve escrever a MH final e o conceito obtido (0, se MH < 6.0; 1 se 6.0 <= MH < 7.0; 2 se 7.0 <= MH < 8.0; 3 se 8.0 <= MH < 9.0; 4 se MH >= 9.0).
Dica: No caso do aluno ter obtido média inferior a 6.0, uma forma de verificar se ele pode alcançar média 6.0 através da prova de recuperação é substituindo a menor nota por 10 e verificando se a MH resultante é igual ou superior a 6.0.

C00000660 - Faça um algoritmo que leia 4 valores, Hi, Mi, Hf, Mf, representando respectivamente a hora e minuto inicial e final de um evento, e calcule a duração do mesmo em horas e minutos. Considere que o evento pode iniciar em um dia e terminar no dia seguinte.

C00000700 - As tarifas de um estacionamento são definidas assim: A primeira e a segunda hora custam 5 reais cada. A terceira e a quarta hora custam 2 reais cada. A partir da quinta hora, cada hora custa 1 real cada. Assim, se um carro ficar 5 horas no estacionamento, o motorista pagará 15 reais (5+5+2+2+1). Faça um algoritmo que leia dois valores He e Hs, respectivamente a hora de entrada e saída no estacionamento (horas inteiras, sem minutos), e escreva o valor a ser pago. Considere que o

C00000750 - Faça um algoritmo que leia, para duas barras de ouro, o seu peso e seu valor. O algoritmo deve ler também o limite de peso de uma mochila, e verificar e escrever que barra (s) devem ir na mochila de modo a maximizar o valor dentro dela, sem exceder seu limite de peso. O algoritmo deve escrever:
0 - Se nenhuma das barras puder ser colocada na mochila sem exceder o limite de peso
1 - Se apenas a barra 1 puder ir na mochila
2 - Se apenas a barra 2 puder ir na mochila
3 - Se ambas as barras puderem ir na mochila simultaneamente

C00000800 - Faça um algoritmo que leia 4(quatro) valores e escreva os 3 (tres) maiores em ordem decrescente. Considere que podem ocorrer valores iguais.

C00000810 - O número 3025 tem a seguinte característica: $30+25=55$ e $55^2=3025$. Faça um algoritmo que leia um número de quatro dígitos e verifique se tem essa característica, escrevendo:
0 - se não tem essa característica;
1 - se tem essa característica.

C00000850 - Faça um algoritmo que leia as três dimensões (largura, comprimento e altura) de duas caixas, e verifique se a primeira caixa pode ser colocada dentro da segunda, escrevendo 1 se é possível colocá-la, e 0 se não é possível. Considere que as caixas não podem ser rotacionadas em nenhuma direção.

Dica: Para que uma caixa caiba na outra, cada dimensão dela deve ser menor (não pode ser igual ou maior) que a dimensão correspondente na segunda caixa

C00000860 - Faça um algoritmo que leia as três dimensões (largura, comprimento e altura) de duas caixas, e verifique se a primeira caixa pode ser colocada dentro da segunda, escrevendo 1 se é possível colocá-la, e 0 se não é possível. Considere que as caixas podem ser rotacionadas em qualquer direção.

C00000865 - Uma fábrica produz um recipiente de plástico com sua tampa (também de plástico). Ambos os componentes utilizam o mesmo equipamento para fabricação (ou seja, não podem ser fabricados ao mesmo tempo). A fabricação do recipiente consome duas horas; a fabricação a tampa consome meia hora. Um cliente deseja o máximo de recipientes (com tampa) para 10 dias. A fábrica trabalha 24 horas/dia e já dispõe de uma quantidade r de recipientes e t de tampas em seu estoque (não necessariamente iguais). Faça um algoritmo que leia os valores de r e t e informe o máximo de conjuntos recipiente-tampa que ela pode fornecer em 10 dias. Dica: Considere 3 casos: os números de recipientes e de tampas no estoque são iguais; os números de recipientes e de tampas no estoque são diferentes e HÁ tempo para igualá-los; os números de recipientes e de tampas no estoque são diferentes e NÃO HÁ tempo para igualá-los (para as duas últimas situações, deve-se analisar separadamente o que há a mais).

C00000866 - Faça um algoritmo que leia a data de nascimento de uma pessoa e a data atual (cada data com dia, mês e ano). O algoritmo deve escrever 1 se a pessoa é maior de idade, e 0 se ela é menor de idade (maioridade: 18 anos).

C00000900 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele entra e sai no mesmo dia.

C00001000 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele entra e sai no mesmo dia, e que as horas a partir das 20:00 valem 20% a mais (adicional noturno).

C00001100 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele entra e sai no mesmo dia, e que as horas antes das 6:00 da manhã e a partir das 20:00 valem 20% a mais (adicional noturno).

C00001200 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele pode entrar em um dia e sair no outro, mas que o total de horas trabalhadas não excede 23 horas.

C00001250 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele pode entrar em um dia e sair no dia seguinte, e que se ele permanecer mais do que 8 horas, as duas horas a partir da nona hora valem 20% a mais, e as horas a partir da décima primeira hora valem 50% a mais (horas extras).

C00001300 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele pode entrar em um dia e sair no outro, mas que o total de horas trabalhadas não excede 23 horas. Considere que as horas a partir das 20:00 valem 20% a mais (adicional noturno).

C00001400 - Faça um algoritmo que leia para um trabalhador o valor que ganha por hora, a hora de entrada e a hora de saída (valores inteiros, sem minutos) e calcule quanto ele ganhou pelo turno. Considere que ele pode entrar em um dia e sair no outro, mas que o total de horas trabalhadas não excede 23 horas.

Considere também que as horas antes das 6:00 da manhã e a partir das 20:00 valem 20% a mais (adicional noturno).

C00001500 - Faça um algoritmo que leia 5 valores entre 1 e 6, já ordenados, correspondente ao arremesso de 5 dados no jogo do general, e escreva:

- 1 - Se os 5 valores são iguais
- 2 - Se há 4 valores iguais e um diferente
- 3 - Se os 5 valores formam uma sequência (1,2,3,4,5 ou 2,3,4,5,6)
- 4 - Se os valores formam um full-hand (3 valores iguais entre si, e os outros dois valores também iguais entre si)
- 5 - Nenhuma das combinações acima

C00001550 - Faça um algoritmo que leia três valores Dia, Mês e Ano, e verifique se formam uma data válida. O algoritmo deve escrever 1 se a data for válida, 0 em caso contrário. Considere que ano bissexto é aquele divisível por 4 e que abril, junho, setembro e novembro tem 30 dias, fevereiro tem 28 (29 em ano bissexto) e todos os outros meses tem 31 dias.

C00001600 - Faça um algoritmo que leia a data de nascimento de uma pessoa e a data atual, cada uma com dia, mês e ano, e escreva quantos anos completos a pessoa tem.

C00001700 - Escreva um algoritmo que leia duas datas, cada uma composta de Dia, Mês e Ano, e as escreva em ordem cronológica crescente. Ex: se as datas são 01/04/2000 e 17/05/1988, o algoritmo deve escrever 17/05/1988 01/04/2000.

C00001750 - Escreva um algoritmo que leia três datas, cada uma composta de Dia, Mês e Ano, e as escreva em ordem cronológica crescente. Ex: se as datas são 01/04/2000, 17/05/1988 e 23/10/1969, o algoritmo deve escrever 23/10/1969 17/05/1988 01/04/2000.

C00001790 - Escreva um algoritmo que leia uma data, composta por dia, mês e ano, e verifique se a data corresponde ao último dia do mês, escrevendo 1, se for o último dia do mês, e 0 se não for o último dia do mês. Considere, para simplificar o problema, que ano bissexto é aquele divisível por 4, e que fevereiro tem 28 dias (29 em ano bissexto), setembro, abril, junho e novembro têm 30 dias e todos os outros meses tem 31 dias.

C00001800 - Escreva um algoritmo que leia uma data, composta por dia, mês e ano, e escreva a data correspondente ao dia seguinte. Considere, para simplificar o problema, que ano bissexto é aquele divisível por 4, e que fevereiro tem 28 dias (29 em ano bissexto), setembro, abril, junho e novembro têm 30 dias e todos os outros meses tem 31 dias.

C00001850 - Escreva um algoritmo que leia uma data, composta por dia, mês e ano, e escreva quantos dias passaram-se desde o início do ano. Considere, para simplificar o problema, que ano bissexto é aquele divisível por 4, e que fevereiro tem 28 dias (29 em ano bissexto), setembro, abril, junho e novembro têm 30 dias e todos os outros meses tem 31 dias.

C00001900 - Para enviar uma carta são necessários um selo e um envelope. O selo custa 12 centavos e o envelope custa 5 centavos. Faça um algoritmo que leia uma quantia inicial de selos, envelopes e centavos, e escreva o número de cartas que podem ser enviadas com esses selos, envelopes e centavos (utilizando-os para comprar mais selos e envelopes). Considere que não é possível converter selos ou envelopes em dinheiro.

C00001925 - Para enviar uma carta são necessários um selo e um envelope. O selo custa 12 centavos e o envelope custa 5 centavos. Faça um algoritmo que leia uma quantia inicial de selos, envelopes e dinheiro (em reais), e escreva o número de cartas que podem ser enviadas com esses selos, envelopes e centavos (utilizando-os para comprar mais selos e envelopes). Escreva também, nessa ordem, a quantidade de selos, envelopes e dinheiro (em centavos), que restará após

enviadas as cartas. Considere que não é possível converter selos ou envelopes em dinheiro.

C00001950 - Uma fábrica produz um recipiente de plástico com sua tampa (também de plástico). Ambos os componentes utilizam o mesmo equipamento para fabricação (ou seja, não podem ser fabricados ao mesmo tempo). A fabricação do recipiente consome duas horas; a fabricação da tampa consome meia hora. Um cliente deseja o máximo de recipientes (com tampa) para 10 dias. A fábrica trabalha 24 horas/dia e já dispõe de uma quantidade r de recipientes e t de tampas em seu estoque (não necessariamente iguais). Faça um algoritmo que leia os valores de r e t e informe o máximo de conjuntos recipiente-tampa que ela pode fornecer em 10 dias.

C00002000 - Em uma empresa existem 5 faixas de horários 0-8 8-12 12-14 14-18 18-24, nas faixas 0-8 12-14 18-24 o valor pago por hora trabalhada é dobrado. Faça um algoritmo que leia o valor ganho por hora pelo funcionário, a hora de entrada e a hora de saída, e calcule e escreva o salário deste empregado. Considere que ele entra e sai no mesmo dia.

C00002001 - Em uma empresa existem 5 faixas de horários 0-8 8-12 12-14 14-18 18-24, nas faixas 0-8 12-14 18-24 o valor pago por hora trabalhada é dobrado. Faça um algoritmo que leia o valor que o funcionário ganha por hora, a hora de entrada e a hora de saída, calcule e escreva o salário deste empregado. (Pode entrar em um dia e sair no outro, mas não pode trabalhar mais de 23 horas)

C00002100 - A distância entre dois pontos definidos pelas coordenadas (X_1, Y_1) e (X_2, Y_2) é dada por $\text{raizq}((X_1 - X_2)^2 + (Y_1 - Y_2)^2)$. Faça um algoritmo que leia 8 valores representando as coordenadas X e Y de 4 pontos e verifique se os pontos formam um quadrado, escrevendo:

1 - se formam um quadrado;
0 - se não formam.

Considere que os pontos são lidos no sentido horário, seguindo o perímetro do quadrado.

C00002150 - Faça um algoritmo que leia oito valores correspondentes às coordenadas dos quatro vértices de um quadrilátero convexo no espaço cartesiano $(X_0, Y_0, X_1, Y_1, X_2, Y_2, X_3, Y_3)$. O algoritmo deve identificar o tipo de polígono formado escrevendo: 1 - se os 4 pontos formam um quadrado. 2 - se formam um retângulo; 3 - se formam um losango; 4 - se formam um paralelograma; 5 - se formam um papagaio (2 pares de lados adjacentes iguais. Ex: lados de tamanhos 3,3,4 e 4); 6 - se não formam nenhum dos anteriores. A distância (tamanho dos lados) entre dois pontos quaisquer (X_i, Y_i) e (X_j, Y_j) pode ser obtida através da fórmula $\text{dist}X_iY_iX_jY_j = \text{raizq}((X_i - X_j)^2 + (Y_i - Y_j)^2)$. Por exemplo, se os pontos lidos foram (3,2), (0,5), (3,8) e (6,5), a figura formada é um quadrado e o algoritmo escreve 1. Para que a figura seja um quadrado, os comprimentos das duas diagonais devem ser iguais, bem como os 4 lados. Se os pontos lidos foram (4,2), (1,4), (4,6) e (7,4), a figura formada é um losango. Se os pontos lidos foram (2,3), (0,5), (3,8) e (5,6), a figura formada é um retângulo. Se os pontos lidos foram (7,3), (0,3), (2,5) e (5,5), a figura formada não é nenhuma das anteriores e o algoritmo escreve 6 (ao menos um lado é diferente dos demais). Os pontos são fornecidos em sentido horário ao longo do perímetro do quadrilátero.

C00003000 - Escreva um algoritmo que leia dois valores D e DS , correspondentes a um dia do mês, e ao dia da semana que corresponde a ele (1-domingo 2-segunda 3-terça 4-quarta 5-quinta 6-sexta 7-sábado). Calcule e escreva em que dia da semana caiu o dia primeiro do mês do dia digitado.
Exemplo: dia 10 no mês e 3 (terça) na semana. Resposta 1 (domingo)
Exemplo: dia 30 no mês e 4 (quarta) na semana. Resposta 3 (terça-feira)

C00003100 - Faça um algoritmo que leia as dimensões (altura, largura e profundidade) de duas caixas e verifique se a primeira caixa pode ser colocada

dentro da segunda. Considere que as caixas podem ser rotacionadas em qualquer direção. Se a primeira caixa couber dentro da segunda escreva 1, caso contrário escreva 0.

C00003200 - Faça um algoritmo que leia dois números de três dígitos cada um, e verifique se possuem os mesmos dígitos. Considere que pode haver dígitos repetidos em um número, e que a cada dígito de um número deve corresponder exatamente um dígito do outro número. Assim, os números 221 e 112 não tem os mesmos dígitos, porque apesar de ambos terem somente os dígitos 1 e 2, aos dois dígitos 2 do primeiro número corresponde o mesmo dígito no segundo número. O algoritmo deve escrever 1, caso os números tenham os mesmos dígitos, e 0 em caso contrário.

C00003250 - Faça um algoritmo que leia um número de três dígitos e escreva o maior número que possui os mesmos dígitos do número lido. Se um dígito aparecer repetido no número lido, deve ser repetido o mesmo número de vezes no número gerado.

C00003300 - Faça um algoritmo que leia 4 valores X_1, Y_1, X_2, Y_2 , correspondendo às coordenadas do canto inferior esquerdo e canto superior direito de uma região retangular no plano. Leia a seguir dois valores X, Y correspondendo a um ponto no plano e escreva: 0 - Se o ponto está fora da região retangular; 1 - Se o ponto está dentro da região retangular; 2 - Se o ponto está exatamente na borda da região retangular.

C00003400 - Faça um algoritmo que leia um número de 8 dígitos e verifique se ele é palíndromo. Ou seja, se a leitura da direita para a esquerda é igual a leitura da esquerda para a direita. Escrever 0 se for palíndromo e 1 se não for.

C00003500 - Um número é primo se divisível somente por ele mesmo e por 1. Faça um algoritmo que leia um número inteiro entre 2 e 100 e determine se ele é ou não um número primo escrevendo 1, se for primo, e 0 em caso contrário.

C00003600 - Faça um algoritmo que leia 6 valores inteiros e escreva ao final a quantidade de valores ímpares lidos, e a soma dos valores pares lidos.

C0003650 - A locadora Só Filmassos classifica os filmes em "lançamentos" e "acervo", sendo que a diária do lançamento é 5 reais e a diária do acervo é 3 reais. A locadora está com uma promoção, ao locar 2 lançamentos o cliente tem direito a um filme de acervo grátis. Faça um algoritmo que leia a quantidade de lançamentos e a quantidade de filmes de acervo a serem locados, e calcule o valor (mínimo) a ser pago.

C00003700 - A locadora Só Filmassos classifica os filmes em "lançamentos" e "acervo", sendo que a diária do lançamento é 5 reais e a diária do acervo é 3 reais. A locadora está com uma promoção, ao locar 2 lançamentos o cliente tem direito a um filme grátis, podendo ser de acervo ou lançamento. Faça um algoritmo que leia a quantidade de lançamentos e a quantidade de filmes de acervo a serem locados, e calcule o valor (mínimo) a ser pago.

Iterativos

I00000100 - Faça um algoritmo que escreva os números de 1 a 20.

I00000200 - Faça um algoritmo que escreva todos os números pares entre 1 e 50.

I00000300 - Faça um algoritmo que escreva todos os números entre 1 e 200 que são múltiplos de 11.

I00000400 - Faça um algoritmo que leia 10 números e escreva os que forem pares. Dica: Um número inteiro é par se o resto da sua divisão inteira por 2 é igual a 0.

I00000500 - Faça um algoritmo que leia números até que seja digitado um número

negativo, e escreva todos que forem pares.

I00000600 - Faça um algoritmo que leia 30 valores e conte quantos estão no intervalo [10,20], escrevendo ao final essa informação.

I00000700

Faça um algoritmo que leia 30 valores e conte quantos estão em cada um dos intervalos [0,25], (25,50], (50,75], (75..100], escrevendo ao final essa informação.

I00000750 - Faça um algoritmo que leia um valor N. Escreva a seguir os números de 1 a N.

I00000800 - Faça um algoritmo que leia um valor N. Leia, a seguir, N valores e escreva todos que forem positivos.

I00000900 - Faça um algoritmo que leia um valor N, e escreva todos os seus divisores.

I00001000 - Faça um algoritmo que leia um valor N, e conte quantos divisores possui, escrevendo ao final essa informação.

I00001100 - Faça um algoritmo que leia um valor N, e calcule e escreva a soma dos seus divisores.

I00001150 - Faça um algoritmo que leia um valor N inteiro e maior do que 1, e calcule e escreva o seu maior divisor (excetuando N).

I00001155 - Faça um algoritmo que leia um valor N inteiro e maior do que 1, e calcule e escreva o seu menor divisor maior do que 1.

I00001200 - Um número inteiro maior do que 1 é primo se ele possui como divisores somente o 1 e ele mesmo. Faça um algoritmo que leia um número e verifique se é primo escrevendo: 1 - se o número é primo; 0 - se o número não é primo. Dica:Pode-se também verificar se um número é primo encontrando seu primeiro divisor maior que 1. Se o primeiro divisor for o próprio número, ele é primo.

I00001300 - Dois números n1 e n2 são ditos amigos entre si se a soma dos divisores de n1 (excluindo o próprio n1) é igual a n2, e a soma dos divisores de n2 (excluindo o próprio n2) é igual a n1. Ex: 220 e 284. Faça um algoritmo que leia 2 valores e verifique se são amigos entre si escrevendo: 1 - se são amigos; 0 - se não são amigos.

I00001400 - Faça um algoritmo que leia 10 números positivos e escreva ao final o maior deles.

I00001500 - Faça um algoritmo que leia 10 números, positivos ou negativos, e escreva ao final o maior deles.

I00001600 - Faça um algoritmo que leia 10 números e escreva ao final o maior e o menor deles.

I00001610 - Faça um algoritmo que leia um valor n ($n \geq 1$) correspondente ao número de alunos de uma turma. Após, o algoritmo lê as notas das provas dos n alunos dessa turma. As notas deverão ser lidas até que sejam informadas n notas válidas, ou seja, no intervalo [0, 10], descartando as notas fora desse intervalo. As notas somente poderão ser lidas uma única vez. O algoritmo deve informar qual foi a menor nota e o percentual dos alunos que tiraram a menor nota (que não é, necessariamente, 0). Por exemplo, se o valor lido para n foi 20 e as notas foram 6.0 6.5 8.0 9.0 4.5 3.0 9.0 8.5 4.5 3.0 6.0 3.0 8.0 9.0 4.5 10 9.0 8.5 4.5 3.0 o algoritmo escreve 3.0 e 20, já que quatro (20% de 20) alunos

tiraram essa nota.

I00001700 - Faça um algoritmo que leia, para 10 pessoas, seu peso e altura e escreva o peso e a altura da pessoa mais alta.

I00001750 - O índice de massa corporal (IMC) de uma pessoa é dada pela relação $\text{peso}/(\text{altura}^2)$. Faça um algoritmo que leia, para 10 pessoas, seu peso e altura e escreva o IMC, peso e altura da pessoa de maior IMC.

I00001800 - Faça um algoritmo que leia o código e a votação de cada um dos candidatos a uma vaga no senado, até que o código digitado seja 0 (não deverá ser lida a votação neste caso). O algoritmo deve escrever os códigos e as votações dos dois senadores eleitos (assuma que não haverão três candidatos com igual votação e que seja a maior votação).

I00001900 - Faça um algoritmo que leia 10 números e escreva os que forem primos.

I00001950 - Faça um algoritmo que escreva os 50 primeiros números primos.

I00002000 - Faça um algoritmo que leia 2 números N1 e N2 e escreva a soma dos números primos entre N1 e N2 (incluindo N1 e N2 se algum deles for primo)..

I00002100 - Faça um algoritmo que leia 2 números N1 e N2 e escreva o produto dos números primos entre N1 e N2 (incluindo N1 e N2 se algum deles for primo).

I00002200 - Faça um algoritmo que leia um número N e escreva os N primeiros números primos maiores que 100.

I00002250 - Faça um algoritmo que leia um número inteiro N e escreva o maior número primo menor do que N.

I00002260 - Faça um algoritmo que leia um número inteiro N e escreva o menor número primo maior do que N.

I00002270 - Um número primo é um número natural maior que 1, que é divisível somente por 1 e por ele mesmo. Faça um algoritmo que leia um número inteiro N e escreva o número mais próximo a ele. Se N for primo, considere que o mais próximo é o próprio N. Se houver dois números à mesma distância, escreva os dois em ordem crescente.

I00002280 - A conjectura de Goldbach diz que todo número par maior que 2 pode ser representado como a soma de dois números primos. Assim, $4=2+2$, $6=3+3$, $8=3+5$... Faça um algoritmo que leia um número N, par, e escreva, em ordem crescente, os dois números primos que o compõem. No caso de haver mais de um par de números (p.ex: $20=3+17$ e $20=7+13$) escreva o par que tiver o menor número primo.

I00002300 - Um número perfeito é o número que é igual à soma de seus divisores, exceto o próprio número (ex: $6 = 1 + 2 + 3$). Faça um algoritmo que leia 10 números e verifique para cada um se é perfeito ou não, escrevendo: 1 - se o número é perfeito; 0 - se o número não é perfeito.

I00002350 - Faça um algoritmo que leia, para um funcionários: o valor que ganha por hora e 30 pares de valores (hora de entrada e hora de saída, inteiros, sem minutos) e calcule o quanto ganhou no mês. O funcionário não pode trabalhar mais de 23 horas seguidas e pode iniciar em um dia e terminar no dia seguinte. Para o funcionário deve ser escrito o quanto ganhou no mês.

I00002400 - Faça um algoritmo que leia, para um número não determinado de funcionários, seu código (inteiro), o valor que ganha por hora e 30 pares de valores (hora de entrada e hora de saída, inteiros, sem minutos) e calcule o quanto ganhou no mês. O funcionário não pode trabalhar mais de 23 horas seguidas e pode iniciar em um dia e terminar no dia seguinte. Para cada funcionário deve

ser escrito seu código e o quanto ganhou no mês. O algoritmo termina ao ser digitado um código igual a -1 (após o que, não deve ser lido mais nenhum dado).

I00002500 - A série de fibonacci é uma sequência de números que inicia por 0 e 1, e a partir do terceiro valor cada número é dado pela soma dos dois anteriores. Assim, os 10 primeiros termos da série são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. Faça um algoritmo que escreva os 100 primeiros termos da série de fibonacci.

I00002525 - A sequência de Fibonacci é formada pelos números: 0 1 1 2 3 5 8 13 21 34 55 ...; sendo que
fibonacci(0)=0; fibonacci(1)=1; fibonacci(2)=1;
fibonacci(3)=2; fibonacci(4)=3; fibonacci(10)=55; ...
Faça um algoritmo que leia um número (N) e escreva o enésimo número da série de Fibonacci.

I00002550 - A sequência de fibonacci é uma sequência de números que inicia por 0 e 1, e a partir do terceiro valor cada número é dado pela soma dos dois anteriores. Assim, os 10 primeiros termos da série são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. Faça um algoritmo que escreva entre os 20 primeiros termos da série de fibonacci apenas os que são primos.

I00002600 - O MDC (máximo divisor comum) entre dois números n1 e n2 é o maior número que é divisor de n1 e de n2. Faça um algoritmo que leia dois números e escreva seu MDC.

I00002700 - O fatorial de um número N (representado por N!) é o produto de todos os números de 1 a N. Assim, $4! = 1 \times 2 \times 3 \times 4 = 24$. Faça um algoritmo que leia um número N e escreva seu fatorial.

I00002750 - O número de combinações de N diferentes objetos em grupos de P objetos, é dado por $N!/(P!(N-P)!)$. Faça um algoritmo que leia uma quantidade N de objetos e o tamanho P dos grupos a serem formados, e calcule a quantidade de grupos que podem ser formados.

I00002800 - Faça um algoritmo que leia 10 valores inteiros menores que 20 e, para cada um, calcule e escreva seu fatorial. O algoritmo deve ignorar todos os valores maiores ou iguais a 20.

I00002900 - O MMC (mínimo múltiplo comum) entre dois números n1 e n2 é o menor número que é múltiplo de n1 e de n2. Faça um algoritmo que leia dois números e escreva seu MMC.

I00003000 - Faça um algoritmo que leia um número N e efetue sua fatoraÇÃO, escrevendo os fatores que o compõem. Ex: $28 = 2 \times 2 \times 7$ $60 = 2 \times 2 \times 3 \times 5$

I00003050 - Faça um algoritmo que leia um número e escreva seus dígitos na ordem contrária em que aparecem no número. Por exemplo, se o número lido foi 32417, o algoritmo escreve 7,1,4,2,3.

I00003100 - Faça um algoritmo que leia 10 números e para cada um escreva a soma dos seus dígitos formantes.

I00003200 - Faça um algoritmo que leia um número N e o escreva com seus dígitos invertidos.

I00003300 - Faça um algoritmo que leia um número N até (até 9 dígitos), retire os dígitos ímpares, e escreva o número resultante. Caso o número só possua dígitos ímpares, o algoritmo não deve escrever nada.

I00003350 - Faça um algoritmo que leia números até que seja fornecido um número negativo. O algoritmo deve escrever a média aritmética dos 3 últimos números fornecidos (excetuando o negativo indicando fim da sequência).

I00003400 - Faça um algoritmo que leia um número N até 100.000.000 e verifique se é palíndromo, ou seja, se se igual quando lido da esquerda para a direita e da direita para a esquerda. Ex: 13731. Escreva 1 se for palíndromo e 0 se não for.

I00003500 - Faça um algoritmo que escreva os primeiros 20 números cuja soma dos dígitos formantes é 10.

I00003600 - Uma pessoa aplicou um determinado valor em um banco. Sabe-se que o banco pagará 5% ao mês de juros. Fazer um algoritmo que leia o valor aplicado e calcule e escreva a quantidade mínima de meses necessários para que a pessoa obtenha R\$1000,00 ou mais de rendimento. Por exemplo, se a pessoa aplicou 10.000 reais, ao final do primeiro mês terá

I00003700 - Faça um algoritmo que leia 20 números inteiros e escreva quantos números são iguais ao menor número lido. Dica: Use um contador, incrementando-o ao encontrar um elemento igual ao menor corrente, e reiniciando-o ao encontrar um elemento menor do que o menor corrente.

I00003800 - Data juliana é o número de dias transcorridos no ano (ex:236/1995). A faixa é de 1 a 365 (366 se o ano for bissexto). Escreva um algoritmo que leia uma data juliana (dia e ano) e a converta para o formato DD/MM/AAAA escrevendo a data. Considere que ano bissexto é aquele divisível por 4.

I00003900 - Faça um algoritmo que leia duas datas, cada uma formada por dia, mês e ano, e escreva o número de dias entre as duas, incluindo a data inicial e a data final. Considere que ano bissexto é aquele divisível por 4.

I00003950 - Faça um algoritmo que leia duas datas, cada uma formada por dia, mês e ano, e escreva todas as datas entre as duas, incluindo a data inicial e a data final. Considere que ano bissexto é aquele divisível por 4. Considere também que a data inicial é menor ou igual à data final.

I00003960 - Faça um algoritmo que leia 4 valores d,m,a,ds, onde d,m,a representam uma data (dia,mês,ano) e ds representa um dia da semana (1-domingo, 2-segunda, 3-terça, 4-quarta, 5-quinta, 6-sexta, 7-sábado), e escreva as datas das próximas 3 sextas-feiras 13, a partir da data digitada, incluindo a própria data, se for o caso. Considere que ano bissexto é o ano divisível por 4.

I00003970 - Faça um algoritmo que leia o mês e o ano de uma sexta-feira 13 e escreva o mês e ano das próximas 5 ocorrências de sexta-feira 13. Considere que ano bissexto é o ano divisível por 4.

I00004000 - Faça um algoritmo que calcule e escreva a soma dos 100 primeiros termos da sequência a seguir: $1+3+5+7+9+\dots$

I00004100 (não está no portal porque o resultado é real)
Faça um algoritmo que calcule e escreva a soma dos 100 primeiros termos da sequência a seguir: $1/1 + 1/3 + 1/5 + 1/7 + 1/9 + \dots$

I00004200 (não está no portal porque o resultado é real)
Faça um algoritmo que leia um valor X e calcule e escreva a soma dos 100 primeiros termos da sequência a seguir: $X/1 + X/3 + X/5 + X/7 + X/9 + \dots$

I00004300 (não está no portal porque o resultado é real)
Faça um algoritmo que leia um valor X e calcule e escreva a soma dos 100 primeiros termos da sequência a seguir: $X/1 - X/3 + X/5 - X/7 + X/9 - \dots$

I00004400 (não está no portal porque o resultado é real)
Faça um algoritmo que leia um valor N e escreva a soma dos N primeiros termos da série a seguir:
 $1/2 + 1/3 + 1/5 + 1/7 + 1/11 + 1/13 + 1/17 + \dots$ ou seja, a série onde os denominadores são os números primos

I00004500 - Um número piramidal é um número que é igual à soma de 3 números primos consecutivos (ex: $15 = 3 + 5 + 7$). Faça um algoritmo que leia um valor N e escreva os 10 primeiros números piramidais maiores ou iguais a N.

I00004600 - Faça um algoritmo que leia 30 números. O algoritmo deve escrever ao final a média dos números pares digitados (soma dos pares dividido pela quantidade de pares); quantos números são primos e o produto (multiplicação) de todos os números lidos que são múltiplos de 5.

I00004700 - Faça um algoritmo que leia 20 números e verifique qual a maior sequência estritamente crescente (isto é, cada número é maior que o anterior) dentre os números digitados, escrevendo ao final o primeiro e último valor da sequência.

I00004800 - Faça um algoritmo que leia uma sequência de 20 números e escreva: 0 - Se todos os números são iguais; 1 - Se a sequência é não-decrescente (cada número é maior OU IGUAL ao anterior); 2 - se a sequência é estritamente crescente (cada número é MAIOR que o anterior); 3 - Se a sequência é não-crescente (cada número é menor OU IGUAL ao anterior); 4 - Se a sequência é estritamente decrescente (cada número é MENOR que o anterior); 5 - A sequência é desordenada (há partes crescentes e partes decrescentes)

I00004850 - Faça um algoritmo que leia 15 números e identifique a maior sequência de números pares dentre os números lidos, escrevendo ao final o comprimento da sequência identificada, bem como o primeiro e o último valor da sequência. No caso de mais de uma sequência do mesmo comprimento, o algoritmo deve considerar a primeira delas.

I00004900 - Faça um algoritmo que leia a média e a quantidade de faltas para cada aluno de um grupo de 20 alunos, e escreva ao final o percentual de alunos aprovados. Os alunos serão considerados reprovados se não atingirem média 6.0. Também serão reprovados os alunos que tiverem mais de 20 faltas. Os alunos com uma quantidade de faltas entre 10 e 20 serão aprovados se obtiverem uma média mínima de 7.5.

I00005000 - Faça um algoritmo que leia um número inteiro qualquer e escreva os dígitos desse número em ordem crescente do valor de cada dígito.

I00005100 - Faça um algoritmo que leia um número inteiro qualquer e verifique se possui algum dígito repetido escrevendo ao final: 0 - se ele não contém nenhum dígito repetido; 1 - se ele contém algum dígito repetido.

I0005120 - Faça um algoritmo que leia um número inteiro e escreva quantas vezes ocorre o dígito 2.

I00005150 - Faça um algoritmo que leia um número inteiro (máximo 5 casas - não maior que 99999) e mostre quantos dígitos contém de cada número de 0 a 9

I00005160 - Faça um algoritmo que leia um número inteiro (máximo 5 casas - não maior que 99999). Mostre o número que mais se repete. Em caso de empate, mostre o menor deles.

I00005170 - Um número inteiro é dito ascendente se cada um dos seus algarismos é maior do que o algarismo imediatamente à sua esquerda. Por exemplo, o número 3589. Faça um algoritmo que leia um número inteiro e verifique se ele é ou não ascendente escrevendo: 1 - se ele é ascendente; 0 - se ele não é ascendente.

I00005200 - Faça um algoritmo que leia o dia e mês de uma data, e o dia da semana (1 - domingo, 2 - segunda... 7 - sábado) e escreva o dia da semana correspondente ao dia primeiro do mês digitado.

I00005300 - Faça um algoritmo que leia um número de 8 dígitos e calcule a soma ponderada dos seus dígitos, com peso 1 para o dígito menos significativo, peso 2 para o segundo dígito, peso 3 para o terceiro e assim por diante. Escreva a soma

calculada.

I00005400 - Faça um algoritmo que leia 20 números e escreva a maior sequência de valores iguais entre os números digitados.

I00005450 - Faça um algoritmo que leia um número inteiro positivo N não maior que 1000000000000000000 (1E18), calcule e escreva o maior inteiro menor ou igual a sua raiz quadrada, ou seja, calcule e escreva a parte inteira da raiz quadrada no número informado.

I00005500

Georg Cantor demonstrou que os números racionais são enumeráveis pela sequência:

```
1  1  2  1  2  3  1  2  3  4  1  2
-----
1  2  1  3  2  1  4  3  2  1  5  4
```

Fazer um algoritmo que leia N (no intervalo de 1 até 1000000) e escreva o enésimo número dessa sequência (escreva o numerador e o denominador).

I00005501

Georg Cantor demonstrou que os números racionais são enumeráveis pela sequência:

```
1  1  2  1  2  3  1  2  3  4  1  2
-----
1  2  1  3  2  1  4  3  2  1  5  4
```

Fazer um algoritmo que leia N (no intervalo de 1 até 1000000000000000000 = 1E18) e escreva o enésimo número dessa sequência (escreva o numerador e o denominador).

I00005600 - O CPF é formado por onze dígitos (999999999-99), dos quais os dois últimos são verificadores (controle), ou seja, a partir dos nove primeiros dígitos pode-se determinar os últimos dois. Considerando o CPF no formato abcdefghi-jk, onde cada letra representa um dígito, pode-se:

- calcular o primeiro dígito verificador (j), da seguinte forma:
 - somar: $10a + 9b + 8c + 7d + 6e + 5f + 4g + 3h + 2i$
 - encontrar o resto da divisão dessa soma por 11.
 - se o resto for igual a zero ou um, o dígito é zero, senão o dígito é onze menos esse resto.
- calcular o segundo dígito verificador (k):
 - somar: $11a + 10b + 9c + 8d + 7e + 6f + 5g + 4h + 3i + 2j$
 - encontrar o resto da divisão dessa soma por 11.
 - se o resto for igual a zero ou um, o dígito é zero, senão o dígito é onze menos esse resto.

Fazer um algoritmo que leia o CPF (somente primeiros nove dígitos) e escreva separadamente os verificadores (dois últimos).

I00005700 - Foram entrevistados 500 alunos de uma universidade. Para cada aluno entrevistado foi registrado o código do curso que ele frequenta (1: engenharia; 2: computação; 3: administração) e sua idade. Faça um algoritmo que processe tais dados fornecendo: (a) número de alunos por curso; (b) número de alunos com idade entre [20 25] anos por curso; e (c) código do curso com a menor média de idade.

Dica:São necessários vários contadores.

I00005800 - O quociente da operação de divisão pode ser obtido subtraindo-se o divisor do dividendo. Da diferença, subtraímos novamente o divisor e assim sucessivamente até que a diferença seja menor do que o divisor. A quantidade de subtrações é o quociente. Assim, por exemplo;

```
-21 / 4 =
21 - 4 = 17 (1)
17 - 4 = 13 (2)
13 - 4 = 9  (3)
9  - 4 = 5  (4)
5  - 4 = 1  (5)
```

Para o exemplo acima, o número de subtrações é 5. Logo, o quociente é -5 (divisor e dividendo têm sinais diferentes). Faça um algoritmo que leia o dividendo (0, positivo ou negativo) e o divisor (positivo ou negativo) e escreva o quociente usando o algoritmo acima. NÃO deve ser usado o operador de divisão. Dica: Registre, de alguma forma, os sinais dos operandos e transforme-os para positivo.

I00005810 - A operação de multiplicação, quando o multiplicador é um número inteiro, nada mais é do que uma sucessão de somas. Assim, por exemplo, $4 \times 5 = 5 + 5 + 5 + 5 = 20$.

Escreva um algoritmo que leia o multiplicando e o multiplicador e, através de uma sucessão de somas, calcule e escreva o produto. O multiplicador é, necessariamente, um número inteiro. Porém, multiplicador e multiplicando podem ser 0 ou negativos.

17 - 4 = 13 (2)

13 - 4 = 9 (3)

9 - 4 = 5 (4)

5 - 4 = 1 (5)

Para o exemplo acima, o número de subtrações é 5. Logo, o quociente é -5 (divisor e dividendo têm sinais diferentes). Faça um algoritmo que leia o dividendo (0, positivo ou negativo) e o divisor (positivo ou negativo) e escreva o quociente usando o algoritmo acima. NÃO deve ser usado o operador de divisão. Dica: Registre, de alguma forma, os sinais dos operandos e transforme-os para positivo.

I00005900 - Um número N é dito um Meridiano Perfeito se existe um número M para o qual a soma dos números de 1 a N-1 é igual à soma dos números de N+1 a M. Assim, 6 é um Meridiano Perfeito porque $1+2+3+4+5=7+8$. Faça um algoritmo que leia um número e verifique se é um meridiano perfeito, escrevendo 1, se o número é um meridiano perfeito, e 0, se não for.

I00006000 - Um número N é dito um Meridiano Perfeito se existe um número M para o qual a soma dos números de 1 a N-1 é igual à soma dos números de N+1 a M. Assim, 6 é um Meridiano Perfeito porque $1+2+3+4+5=7+8$. Faça um algoritmo que leia um valor N, e escreva os N primeiros Meridianos Perfeitos maiores que 1.

I00006100 - Um número é dito regular, ou número de hamming (ou "5-smooth number" ou "ugly number") se tem como fatores primos apenas 2, 3 e 5, ou seja, é divisível somente por múltiplos de 2, 3 ou 5. Assim, os primeiros números regulares são 1 (é regular por definição, $2^0 * 3^0 * 5^0$), 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32.... Faça um algoritmo que leia um número N, menor ou igual a 100, e escreva os N primeiros números regulares.

I00006200 - Faça um algoritmo que para 10 cartas de baralho leia o seu valor (entre 1 e 13) e naipe (1 - ouros, 2 - copas, 3 - paus, 4 - espadas) em sequência, e escreva: 1 - se as cartas contem uma canastra; 0 - se não contem. Considere como canastra apenas uma sequência crescente de 7 cartas do mesmo naipe de numeração contínua (cuja diferença entre duas cartas seja igual a 1). Dica: Considere que podem ocorrer cartas de mesmo número e naipe (uso de mais de um baralho). Nesse caso a carta de mesmo número e naipe não é contada, mas não quebra a sequência.

I00006201 - Faça um algoritmo que leia, para 10 cartas, seu naipe (1 a 4) e número (1 a 13), em ordem decrescente de naipe e número e escreva a maior sequência decrescente (mesmo naipe, numeração decrescente contínua) formada a partir da primeira carta. Caso a sequência possua mais de 7 cartas, escreva apenas as primeiras sete cartas da sequência. Se a sequência tiver menos de 7 cartas, escreva a sequência encontrada e um 0 ao final da sequência.

I00006300 - Faça um algoritmo que leia, para duas cartas, seu valor (1 a 13) e naipe (1 a 4). O algoritmo deve ler, em seguida, a opção de jogo: (1) dar mais uma carta, (0) encerrar jogo. O jogo também acaba, caso a soma das cartas na mão do jogador atinja ou ultrapasse 21. Ao fim do jogo, diga a quantia de cartas que

o jogador recebeu (incluindo as duas cartas iniciais) e qual é a soma das cartas que ele recebeu em mãos caso esta soma não ultrapasse 21. Caso a soma ultrapasse 21, o algoritmo deve escrever o número de cartas e 0

I00006301 - Continuação do exercício <I6300>. Considere que agora as cartas tem numeração contínua de 1 a 52, sendo que as de 1 a 13 são de ouros, de 14 a 26, copas, de 27 a 39, espadas e de 40 a 52, paus. Considere também que as cartas "valetes, rainha e rei", valem uma pontuação igual a 10; E que caso inicialmente o jogador receba um "Ás" e outra carta de valor igual a 10, o Ás valerá 11 e o jogador vence o jogo instantaneamente, e neste caso, deve-se devolver o valor "21" como resposta. Nos demais casos, devolva o número de cartas que ele recebeu, e a soma delas, caso não ultrapasse 21, pois neste caso deve-se devolver "0" como resposta.

I00006302 - Continuação do exercício <I6301>. O motivo pelo qual o jogador recebe vitória automática no caso de receber inicialmente "um Ás e uma carta de valor 10", se deve ao fato de que o "Ás" pode assumir dois valores "1" e "11", de acordo com a situação; Logo um Ás e outra carta de valor 10, equivalem à soma "11+10=21", entretanto, se o jogador possuísse em mãos uma soma desfavorável: Um Ás, um nove e um dez. O Ás assume seu menor valor "1+9+10", pois "11+9+10" geraria derrota. A partir destes dados, construa um algoritmo que integre o jogo de blackjack, averiguando está nova informação.

I00006303 - Continuação do exercício <I6302>. O jogador, nunca joga "sozinho" como talvez você possa ter imaginado, afinal, ele poderia parar com o menor número de cartas e sempre vencer! Por isto, ele tem neste caso, como concorrente, a "casa", porém, esta não joga para "vencer" o jogo, e sim, para simplesmente impedir o jogador de vencer. Funciona da seguinte forma: O jogador faz todas as suas jogadas, até parar, se ele não tiver passado de 21 ou ganhou automaticamente (tirando 21 com as duas primeiras cartas), passa para a "casa", que diferente do jogador, só precisa ter uma quantia "maior" ou "igual" de pontos na mão, pois em caso de "empate" nenhum dos dois ganha. Ao fim, diga, se o jogador venceu(1), se a "casa" venceu(2) ou se houve empate(3). Considere que serão dados inicialmente, o número de pontos que o jogador fez (podendo este ser maior que 21. Neste caso, dê a resposta de vitória para a "casa"), em seguida, virão os dados com os quais a mesa joga. Considere também que as cartas não passam do valor 13.

I00006400 - Baseando-se na sequência de fibonacci (0,1,1,2,3...), verifique se uma sequência de cinco números, faz parte da sequência, devolvendo como resposta "0" se faz parte, ou "1" se não faz parte. Dica: Considere que números "repetidos" fazem parte da sequência proposta, ou seja, devem ser analisados individualmente como os demais números fornecidos.

I00006500 - Leia um termo n que define o tamanho máximo dos lados de um triângulo retângulo podem assumir, sendo que os lados começam com tamanho 3, 4 e 5 (a,b,c). Em sequência escreva todos os triângulos retângulos em que o valor máximo do lado é n e os três lados são inteiros. Cada triângulo deve ser escrito uma única vez em ordem crescente dos lados. Dica: O triângulo retângulo é dado como válido pela fórmula ($a^2 + b^2 = c^2$)

I00006800 - Fazer um algoritmo que leia números inteiros até que seja informado 0, descartando os números negativos. Escrever o menor número com a maior quantidade de dígitos.

Vetores

V00000100 - Escrever um algoritmo que lê um vetor V(10) e o escreve. Conte, a seguir, quantos valores de V são negativos e escreva esta informação.

V00000200 - Faça um algoritmo que leia 10 valores e os escreva na ordem contrária à que foram digitados.

V00000250 - Faça um algoritmo que leia os conceitos (1, 2, 3, 4 ou 5) dos alunos

de uma turma de 20 alunos. Após, o algoritmo desenha um histograma com os conceitos na forma

```
111
2222
3333333333
44
5
```

se houveram três alunos com conceito 1, quatro com conceito 2, dez com conceito 3, dois com conceito 4 e um com conceito 5.

V00000300 - Faça um algoritmo que leia 10 valores e escreva primeiramente todos os números pares digitados e em seguida os números ímpares.

V00000320 - Faça um algoritmo que leia uma quantidade indeterminada de números até que sejam informados 10 números positivos (maiores que zero) e em seguida escreva esses números positivos em ordem contrária à ordem de leitura.

V00000400 - Escrever um algoritmo que lê um vetor V(10) e o escreve. Conte, a seguir, quantos valores de V são primos e escreva esta informação.

V00000500 - Faça um algoritmo que leia dez números inteiros armazenando-os em um vetor. Após, escreva a posição de cada número menor que zero desse vetor.

V00000600 - Escreva um algoritmo que lê um vetor A(10) e escreva a posição de cada elemento primo deste vetor.

V00000700 - Escrever um algoritmo que lê um vetor X(10) e o escreve. Substitua, a seguir, todos os valores negativos de X pelos seu módulo e escreva novamente o vetor X.

V00000800 - Escrever um algoritmo que lê um vetor C(50) e o escreve. Encontre, a seguir, o maior elemento de C e o escreva.

V00000900 - Escrever um algoritmo que lê um vetor N(80) e o escreve. Encontre, a seguir, o menor elemento e a sua posição no vetor N e escreva: "O menor elemento de N é = ... e a sua posição é ...".

V00001000 - Faça um algoritmo que leia 10 valores, calcule sua média e escreva os valores que estão acima da média calculada.

V00001010 - A conjectura de Goldbach diz que todo número par maior que 2 pode ser representado como a soma de dois números primos. Assim, $4=2+2$, $6=3+3$, $8=3+5$... Faça um algoritmo que leia um número N, par, e escreva, em ordem crescente, os dois números primos que o compõem. No caso de haver mais de um par de números (p.ex: $20=3+17$ e $20=7+13$) escreva o par que tiver o menor número primo.

V00001050 - Faça um algoritmo que leia um vetor de 10 elementos. Leia, a seguir, um valor N e verifique se o valor aparece no vetor escrevendo: 0 - se o valor N não aparece no vetor; 1 - se o valor N aparece no vetor

V00001100 - Escrever um algoritmo que lê um vetor N(10) e o escreve. Troque, a seguir, o 1º elemento com o último, o 2º com o penúltimo, etc até o 5º com o 6º e escreva o vetor N assim modificado.

V00001200 - Escrever um algoritmo que lê um vetor N(10) e o escreve. Troque, a seguir, cada elemento que estiver em posição ímpar (o primeiro, o terceiro...) pelo elemento da posição imediatamente a seguir. Escreva o vetor modificado.

V00001250 - Escreva um algoritmo que lê em um vetor de 20 posições números positivos, até que o vetor esteja completo ou que tenha sido fornecido o valor 0 (zero). Após, escreva o vetor. A seguir, o algoritmo deve ler um número positivo

qualquer e, caso ele se encontre no vetor, deve remover todas suas ocorrências, através de um deslocamento para a esquerda dos elementos que encontram-se à direita daquele a ser removido. Escreva o vetor modificado.

V00001300 - Escrever um algoritmo que lê um vetor G(10) contendo o gabarito de uma prova. Leia a seguir, para cada aluno de um conjunto de 5 alunos suas 10 respostas e verifique quantas questões acertou, escrevendo para cada aluno o número de acertos.

V00001400 - Faça um algoritmo que leia 10 valores e verifique se algum dos valores aparece repetido. Escreva 1 se algum valor aparece repetido e 0 se não houver nenhum valor repetido.

V00001450 - Faça um algoritmo que leia um vetor de 10 elementos. Leia, a seguir, 10 valores N e, para cada um, verifique se o valor aparece no vetor escrevendo: 0 - se o valor N não aparece no vetor; 1 - se o valor N aparece no vetor

V00001500 - Escrever um algoritmo que lê um vetor X(10) e, após, leia um vetor Y(10). Crie, a seguir, um terceiro vetor Z com os elementos que aparecem nos dois vetores (intersecção). Os elementos devem aparecer no vetor Z na mesma ordem em que aparecem no vetor X. Considere que não há repetição de valores dentro do mesmo vetor. Escreva o vetor Z (apenas as posições que foram preenchidas).

V00001501 - Escrever um algoritmo que lê um vetor X(10) e, após, leia um vetor Y(10). Crie, a seguir, um terceiro vetor Z com os elementos que aparecem em X ou em Y (união); elementos que aparecem em X e Y simultaneamente devem aparecer apenas uma vez em Z. Os elementos devem aparecer no vetor Z na mesma ordem em que aparecem no vetor X e Y. Considere que não há repetição de valores dentro do mesmo vetor. Escreva o vetor Z (apenas as posições que foram preenchidas).

V00001600 - Faça um algoritmo que leia um vetor V(5), com os valores sorteados em um sorteio de Loto. Leia, a seguir, para um conjunto de 5 apostadores, seus 5 palpites e escreva, para cada um, o número de acertos que teve.

V00001650 - Faça um algoritmo que leia um número N e escreva os N primeiros números primos em ordem decrescente. Considere que N é no máximo igual a 100. Ex: Se N=5, escreva 11,7,5,3,2.

V00001700 - Escrever um algoritmo que lê 2 vetores X(10) e Y(10), e escreva os elementos que aparecem no vetor X e não aparecem no vetor Y (diferença de conjuntos). Escreva os valores na ordem em que eles aparecem no vetor X. Os dois vetores devem ser lidos separadamente (em primeiro lugar, todo o vetor X, após, o vetor Y).

V00001710 - Escrever um algoritmo que lê 3 vetores A[1..10], B[1..10] e C[1..10] e escreva os elementos que estão em A e B (intersecção) mas não estão em C. Escreva os valores na ordem em que eles aparecem no vetor A. Os três vetores devem ser lidos separadamente (em primeiro lugar, todo o vetor A, após, o vetor B e por fim o vetor C).

V00001720 - Escrever um algoritmo que lê 3 vetores A[1..10], B[1..10] e C[1..10] e escreva os elementos que são comuns aos três vetores (intersecção). Escreva os valores na ordem em que eles aparecem no vetor A. Os três vetores devem ser lidos separadamente (em primeiro lugar, todo o vetor A, após, o vetor B e por fim o vetor C).

V00001730 - Escrever um algoritmo que lê 2 vetores A[1..10] e B[1..10] e escreva os elementos que estão somente em A ou somente em B. Escreva os valores na ordem em que eles aparecem no vetor A, e em seguida os que aparecem no vetor B. Os dois vetores devem ser lidos separadamente (em primeiro lugar, todo o vetor A e, após, o vetor B).

V00001790 - Faça um algoritmo que leia valores inteiros entre 1 e 10 até que

seja digitado um valor igual a 0, e escreva, ao final, quais dos valores entre 1 e 10 que não foram digitados nenhuma vez.

V00001800 - Faça um algoritmo que leia um vetor X(10). Compare a seguir cada elemento com o elemento da posição seguinte, trocando-os entre si se o elemento de maior valor estiver antes do elemento de menor valor. O que se pode dizer sobre a ordem dos elementos dentro do vetor após essa operação?

V00001900 - Faça um algoritmo que leia um vetor X(10) e ordene seus elementos em ordem crescente. Escreva o vetor ordenado. Dica: o que acontece se o procedimento descrito no exercício V00001800 for executado repetidamente sobre o vetor?

V00002000 - Faça um algoritmo que leia, para cada pessoa de um conjunto de 10 pessoas, o seu peso (inteiro) e altura (real), e escreva a lista de pesos e alturas em ordem crescente de altura. Os dois dados referentes a cada pessoa devem ser lidos juntos (ou seja, o algoritmo não deve ler todos os pesos e em seguida todas as alturas)

V00002100 - Faça um algoritmo que leia dois vetores V(10) e W(10), ordenando cada um em ordem crescente. Gere, a seguir, um vetor Z(20) com os elementos de V e W, também em ordem crescente. Escreva o vetor Z.

V00002200 - Escrever um algoritmo que gera os 10 primeiros números primos acima de 100 e os armazena em um vetor X(10) escrevendo, no final, o vetor X.

V00002250 - Escrever um algoritmo que leia 20 números inteiros em um vetor e escreva o elemento que aparece MENOS vezes no vetor com o número de ocorrências. Suponha que não haverão dois ou mais números com o mesmo número mínimo de ocorrências.

V00002300 - Escrever um algoritmo que lê um vetor X(20) possivelmente contendo repetições e o escreve. Escreva, a seguir, cada um dos valores que aparecem em X dizendo quantas vezes cada valor aparece em X.

V00002400 - Faça um algoritmo para gerar um vetor com os números primos de 1 a 1000 através do crivo de eratóstenes. Neste método o vetor (inicializado com zeros) é percorrido da segunda posição até o fim e para cada posição n que contém 0, n é um número primo e todas as posições múltiplas de n devem receber 1. Escreva ao final os números primos. Eles são indicados pelas posições do vetor que contêm 0.

V00002500 - Faça um algoritmo que leia 30 valores inteiros entre 1 e 10 e escreva o número de vezes que ocorre cada valor.

V00002600 - Faça um algoritmo que leia 10 números reais armazenando-os em um vetor. Desloque, a seguir, todos os elementos do vetor uma posição para a esquerda (colocando na última posição o elemento que estava na primeira posição). Repita isso até o que o maior elemento esteja ocupando a primeira posição.

V00002700 - Faça um algoritmo que coloque em um vetor os 20 primeiros números primos da série de Fibonacci

V00002800 - Faça um algoritmo que leia uma quantidade indeterminada de números até que sejam digitados 10 números distintos e escreva esses 10 números.

V00002900 - Faça um algoritmo que preencha um vetor com os 20 primeiros números primos e após escreva os elementos desse vetor. Para identificar se um número é primo, verifique se é divisível por algum dos primos já colocados no vetor.

V00003000 - Faça um algoritmo que leia dois vetores V(15) e W(15), contendo, cada um, um número de 15 dígitos, cada dígito ocupando uma posição do vetor, o dígito menos significativo do número ocupando a posição 15, e calcule a soma dos

dois números, colocando o resultado em um vetor S(16). Escreva ao final o vetor S. Os dois vetores de entrada devem ser lidos separadamente, primeiro o vetor V e, após, o vetor W.

V00003050 - Faça um algoritmo que leia dois vetores V(15) e W(15), contendo, cada um, um número de 15 dígitos, cada dígito ocupando uma posição do vetor, e efetue a comparação entre os dois números de quinze dígitos em V e W, escrevendo: -1 se o número em V é menor. 0 (zero) se os números em V e W são iguais. 1, se o número em V é maior. Leia inicialmente os 15 dígitos de V e, após, os 15 dígitos de W.

V00003100 - Faça um algoritmo que leia dois vetores V(15) e W(15), contendo, cada um, um número de 15 dígitos, cada dígito ocupando uma posição do vetor, o dígito menos significativo do número ocupando a posição 15, e calcule a diferença dos dois números (V-W), colocando o resultado em um vetor S(16). Escreva ao final o vetor S. Considere que o número de 15 dígitos no vetor V é maior que o número contido no vetor W.

V00003200 - Faça um algoritmo que leia um vetor V(15) contendo um número de 15 dígitos, e um valor N entre 0 e 9, e calcule o produto (multiplicação) do número em V pelo valor N, colocando o resultado em um vetor P(16). Escreva o vetor P.

V00003250 - Faça um algoritmo que leia, para 5 candidatos a deputado, o seu cadastro. Leia, a seguir, um número não determinado de votos (termina ao ser digitado 0) e escreva ao final o cadastro e número de votos de cada candidato (em ordem crescente de cadastro). Escreva também, ao final, o número de votos nulos (ou seja, votos que não são iguais ao cadastro de nenhum candidato).

V00003300 - Faça um algoritmo que leia, para 20 candidatos a deputado, seu cadastro, partido (inteiro entre 1 e 10) e número de votos, e escreva o total de votos de cada partido. Os 3 dados correspondentes a cada candidato devem ser lidos juntos, antes de passar para o próximo candidato.

V00003350 - Faça um algoritmo que leia, para 20 candidatos a deputado, seu cadastro, partido (inteiro entre 1 e 10) e número de votos, e escreva, para os 10 partidos, o número do partido e o total de votos, em ordem decrescente de votação. Os 3 dados correspondentes a cada candidato devem ser lidos juntos, antes de passar para o próximo candidato.

V00003400 - Faça um algoritmo que leia, para 20 candidatos a deputado, seu cadastro, partido (inteiro entre 1 e 10) e número de votos, e escreva o cadastro, partido e número de votos do candidato mais votado de cada partido, em ordem crescente de partido.

V00003410 - Faça um algoritmo que leia, para 20 candidatos a deputado, seu cadastro, partido (inteiro entre 1 e 10) e número de votos, e escreva o cadastro, partido e número de votos do candidato mais votado de cada partido, em ordem decrescente de votação.

V00003450 - Faça um algoritmo que leia, para 20 candidatos a deputado, seu cadastro, partido (inteiro entre 1 e 10) e número de votos, e escreva o cadastro, partido e número de votos de todos os candidatos, em ordem crescente de partido e, para o mesmo partido, em ordem crescente de número de votos.

V00003500 - Faça um algoritmo que leia 20 números reais. Escreva a quantidade mínima de elementos cuja soma seja maior ou igual à metade da soma total. Dica: Classifique o vetor e some os primeiros (ou últimos) elementos até atingir a metade da soma total.

V00003550 - Faça um algoritmo que leia 10 datas de aniversário, cada uma composta por dia e mes, e escreva as 10 datas em ordem cronológica crescente.

V00003600 - Faça um algoritmo que leia 10 datas, cada uma composta por dia, mês e ano, e escreva as 10 datas em ordem cronológica crescente.

V00003700 - Faça um algoritmo que para 10 cartas de baralho leia o seu valor (entre 1 e 13) e naipe (1 - ouros, 2 - copas, 3 - paus, 4 - espadas), e ordene-as em ordem crescente de naipe e, para cada naipe, em ordem crescente de valor. Escreva a lista de cartas ordenada.

V00003750 - Faça um algoritmo que para 9 cartas de baralho leia o seu valor (entre 1 e 13). O algoritmo deve escrever quantas trincas a "mão" contém. Uma trinca é composta de três cartas com o mesmo valor. O naipe das cartas não importa. Cada carta pode aparecer no máximo em uma trinca.

V00003800 - Faça um algoritmo que para 10 cartas de baralho leia o seu valor (entre 1 e 13) e naipe (1 - ouros, 2 - copas, 3 - paus, 4 - espadas), e escreva: 1 - se as cartas contem uma canastra; 0 - se não contem. Considere como canastra apenas uma sequencia crescente de 7 cartas do mesmo naipe de numeração contínua (cuja diferença entre duas cartas seja igual a 1). Considere que podem ocorrer cartas de mesmo número. Nesse caso a carta de mesmo número não é contada, mas não quebra a sequência.

V00003900 - Faça um algoritmo que leia os 5 números obtidos em um arremesso de 5 dados no jogo de General, e verifique se os números contém um full-hand (3 números iguais, e os outros dois números iguais entre si, mas diferentes dos 3 primeiros. Ex. 3,3,3,2,2), e escreva: 1 - se os números contem um full-hand; 0 - se não contem.

V00004000 - Faça um algoritmo que encontre a maior subsequência de elementos, não necessariamente contíguos, de um vetor de no máximo 100 elementos de tal forma que cada elemento desta subsequência seja menor ou igual ao anterior. Os dados do vetor devem ser lidos até que seja digitado um valor negativo (que deve ser descartado). O algoritmo deve escrever, ao final, o comprimento da maior subsequência encontrada.

V00004100 - Faça um algoritmo que encontre e escreva a maior subsequência de elementos, não necessariamente contíguos, de um vetor de no máximo 100 elementos de tal forma que cada elemento desta subsequência seja menor ou igual ao anterior. Os dados do vetor devem ser lidos até que seja digitado um valor negativo (que deve ser descartado).

V00004150 - Faça um algoritmo que leia um vetor V(10) e um vetor W(3), e verifique e escreva a primeira posição do vetor V a partir da qual são encontrados, de forma contígua, os três elementos do vetor W. Caso o vetor W não ocorra em V, o algoritmo deve escrever -1.

V00004200 - Faça um algoritmo que leia um vetor de inteiros W[30]. Leia a seguir um vetor V[10] e verifique se a sequência de 10 números no vetor V ocorre no vetor W, escrevendo a posição do vetor W onde inicia a sequencia, caso ela exista, ou 0, caso a sequencia em V não exista em W.

V00006000 - Faça um algoritmo que lê os conceitos de uma turma de N alunos, notas inteiras, de 0 a 10, até que seja digitado um valor < 0 (que deve ser descartado) e escreva, para os valores de 0 a 10, o valor e o número de ocorrências do mesmo.

V00006100 - Crie um algoritmo que lê um vetor G(13) representando o resultado de um teste de loteria esportiva, contendo os valores 1 (coluna 2), 2 (coluna do meio) e 4 (coluna 1). A seguir, para cada apostador, o algoritmo lê o n° de seu cartão e suas apostas para cada um dos 13 jogos, sendo válidos os seguintes valores: 1 (coluna 2), 2 (coluna do meio), 3 (coluna 2 e do meio), 4 (coluna 1), 5 (coluna 1 e 2), 6 (coluna 1 e do meio) e 7 (triplo). O algoritmo calcula, para cada apostador, o n° de acertos e escreve o n° do apostador e seu número de acertos. O algoritmo para de ler quando o número do apostador for 0.

V00006200 - Faça um algoritmo que leia valores até que seja digitado um valor negativo e escreva os 10 últimos valores positivos digitados. Considere que serão digitados pelo menos 10 valores positivos. O algoritmo deve escrever os

valores na mesma ordem em que foram digitados pelo usuário.

V00006300 - Crie um algoritmo que lê o número de matrícula de cada aluno e suas respectivas notas em três provas, em uma turma de 10 alunos. Após, calcular a média harmônica (MH) de cada aluno e verificar o conceito obtido (se $MH < 6.0$, conceito é 0. Se $MH < 7.0$, conceito é 1. Se $MH < 8.0$ o conceito é 2. Se $MH < 9.0$ o conceito é 3. Se $MH \geq 9.0$, conceito é 4). Escrever seu número de matrícula e seu conceito. Dica: Lembre que se uma das três notas é 0, a média harmônica é 0.

V00006400 - Na apuração de um processo de eleição as vagas disponíveis são distribuídos entre os partidos proporcionalmente aos votos obtidos pelo partido. As sobras são distribuídas da seguinte forma: enquanto houver sobras, calcula-se para cada partido a razão entre o número de votos e o número de cadeiras já obtidas + 1. Uma das sobras é atribuída ao partido de maior razão. Isso se repete até que não haja mais sobras. Faça um algoritmo que leia, para 10 partidos, de 1 a 10, o seu número de votos. Leia, a seguir, o total de vagas e escreva, ao final, para os partidos de 1 a 10, o número do partido e o número de cadeiras obtidas.

V00006500 - Faça um algoritmo que leia o código de 5 carros com seus respectivos preços. Ao final, escreva o código do carro mais caro com seu respectivo preço.

V00006600 - Escreva um algoritmo que lê um vetor $V[10]$ e escreva, para cada valor diferente que aparece no vetor, o valor e o número de vezes que o mesmo ocorre no vetor. Escreva os valores em ordem crescente de valor.

V00006700 - Faça um algoritmo que leia um vetor $V[10]$ e ao final escreva o elemento que aparece mais vezes neste vetor.

V00006800 - Faça um algoritmo que leia um vetor $V[15]$, já ordenado. Leia, a seguir, um valor N e verifique se ele se encontra no vetor escrevendo a posição em que se encontra. Caso o valor não esteja no vetor, escreva -1. Tente escrever um algoritmo da forma mais eficiente possível.

V00006900 - Faça um algoritmo que leia os 10 primeiros valores de um vetor $V[20]$, valores estes já ordenados em ordem crescente. Leia, a seguir, 10 valores e insira cada um deles na posição correta do vetor V, de modo a mantê-lo ordenado, deslocando em uma posição para a direita todos os valores que vêm após ele. Escreva, ao final, o vetor V (ordenação por inserção).

V00007000 - Na ordenação por contagem, conta-se, para cada elemento do vetor, o número de elementos menores ou iguais a ele. O valor da contagem é a posição que o elemento ocupará no vetor ordenado. Faça um algoritmo que leia um vetor $V[20]$ e ordene-o em ordem crescente, escrevendo o vetor ordenado. Considere que não há repetição de valores no vetor.

V00007100 - Escreva um algoritmo que leia caracteres em dois vetores A[5] e B[5]. Após os dois vetores terem sido lidos, o algoritmo deverá colocar em um terceiro vetor C o resultado da união dos dois primeiros, ou seja, os caracteres que estão contidos em um ou outro vetor. Assuma que em um mesmo vetor não haverá caracteres repetidos, mas um caractere contido em A poderá também estar contido em B. Nesse caso, apenas uma ocorrência é colocada em C.

V00007150 - Uma loja decidiu fazer um sorteio de prêmios para seus clientes. Serão sorteados 10 números, porém, a loja não quer que um mesmo número receba mais do que um prêmio. Faça um algoritmo que leia cada número sorteado até que tenham sido sorteados 10 números diferentes. O algoritmo escreve os 10 números (distintos) sorteados.

V00007160 - Faça um algoritmo que leia números até que tenham sido lidos 5 números distintos ou 20 números. Escreva, ao final, os 5 números distintos em ordem crescente.

V00007200 - Faça um algoritmo que leia um vetor $V[15]$, já ordenado. Leia, a

seguir, um valor N e verifique se ele se encontra no vetor escrevendo a posição em que se encontra. Caso o valor não esteja no vetor, escreva -1. Tente escrever um algoritmo da forma mais eficiente possível. Dica: Compare N com o valor na posição do meio do vetor. Essa comparação dirá se N está na primeira ou na segunda metade do vetor. (pesquisa binária)

V00007300 - Crie um algoritmo que leia um vetor $v[10]$. Leia a seguir um valor N e faça, no vetor V, uma rotação circular para a esquerda (rotate left) de N posições. Na rotação circular para a esquerda, cada elemento é copiado para a posição à esquerda, e o primeiro elemento é copiado para a última posição. Escreva, ao final, o vetor alterado.

V00007400 - Escreva um algoritmo que lê em um vetor o gabarito das 20 questões objetivas de um prova de vestibular com 4 opções: "A", "B", "C" e "D". O algoritmo deve, a seguir, ler, para cada candidato, seu código e as respostas fornecidas para as 20 questões e escrever o número de acertos. O algoritmo pára de ler quando o código do candidato for igual a 0.

V00007500 - Crie um algoritmo que faça uma progressão aritmética de cinco posições, onde deve ser fornecido o primeiro termo da progressão e a razão desta PA.

V00007600 - Faça um algoritmo que leia um número inteiro e escreva os dígitos binários correspondentes a esse número (máximo 64 dígitos binários). Dica: Para converter um número decimal em binário pode-se proceder sucessivas divisões por dois até que o quociente seja igual a zero. O número binário é obtido considerando restos obtidos nas divisões, mas em ordem contrária.

V00007700 - Leia um vetor de 10 posições que contem números positivos e negativos. Em seguida, descubra qual a sequência de valores dentro dele (intervalo de valores contíguos no vetor) que resulta na maior soma, escrevendo ao final a soma encontrada.

V00007800 - A sequência "Look & Say" funciona da seguinte forma: O seu primeiro termo é sempre "1", e a definição do próximo termo a partir do anterior é "O nº de vezes que um número de repete até que a sequência seja quebrada""O número que estava repetindo"...

Veja do seguinte modo:

1º Termo: 1 (Termo inicial em "1")

2º Termo: 11 (Antes tínhamos "1" vez o número "1")

3º Termo: 21 (Antes tínhamos "2" vezes o número "1")

4º Termo: 1211 (Antes tínhamos "1" vez o número "2" e "1" vezes o número "1")

5º Termo: 111221

6º Termo: 312211

...

Elabore um algoritmo que calcule o enésimo termo dessa sequência. Considere que o número é muito grande e você precisará de 1000 casas no vetor que irá aglomerá-lo.

Matrizes

M00000100 - Faça um algoritmo que leia uma matriz $M[1..5, 1..5]$ e calcule e escreva a soma dos elementos da diagonal principal.

M00000150 - Faça um algoritmo que leia uma matriz $M[1..5, 1..5]$ e calcule e escreva a soma dos elementos da diagonal secundária.

M00000200 - Faça um algoritmo que leia uma matriz $M[1..5, 1..5]$ e calcule e escreva a soma dos elementos em toda a região acima da diagonal principal.

M00000300 - Faça um algoritmo que leia uma matriz $M[1..5, 1..5]$ e calcule e escreva a soma dos elementos em toda a região acima da diagonal secundária.

M00000350 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e calcule e escreva a soma dos elementos que estão ao mesmo tempo acima da diagonal principal e da secundária.

M00000400 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e escreva o maior valor existente na matriz, bem como a linha e coluna onde o valor ocorre.

M00000600 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e troque cada elemento da linha 4 com elemento correspondente da linha 2. Escreva a matriz resultante.

M00000700 - Uma matriz identidade é uma matriz que possui 1 em todos os elementos da diagonal principal, e 0 em todas as outras posições. Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e verifique se é uma matriz identidade escrevendo: 1 - Se é uma matriz identidade; 0 - Se não é uma matriz identidade.

M00000800 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e escreva o número da linha que contenha a maior soma de seus elementos. Considere que a matriz só contém valores positivos.

M00000900 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e escreva o número da linha que contenha a maior soma de seus elementos. Considere que a matriz pode conter valores positivos e negativos.

M00001000 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e gere dois vetores SomaLin[1..5] e SomaCol[1..5], com a soma dos elementos de cada linha e a soma dos elementos de cada coluna da matriz M. Escreva ao final os vetores SomaLin e Somacol.

M00001010 - Escrever um algoritmo que lê uma matriz $M[1..3,1..3]$, contendo uma posição de jogo da velha, com valores 0 (casa livre), 1 (marcada com cruzinha) ou 2 (marcada com bolinha) e escreva: 1 - se o jogador 1 venceu o jogo (alguma linha, coluna ou diagonal com o mesmo valor); 2 - se o jogador 2 venceu o jogo; 3 - se o jogo terminou empatado (não há mais lances e ninguém ganhou); 4 - se o jogo ainda não terminou (há lances por jogar e ninguém ainda venceu)

M00001100 - Uma matriz é dita Diagonalmente Dominante se em todas as linhas o elemento da diagonal principal é maior ou igual à soma dos outros elementos da linha, e há pelo menos uma linha em que o elemento da diagonal principal é MAIOR que a soma dos outros elementos da linha (não basta que seja igual). Faça um algoritmo que leia uma matriz $M[1..4,1..4]$ e verifique se é diagonalmente dominante escrevendo: 1 - Se é diagonalmente dominante; 0 - Se não é diagonalmente dominante

M00001200 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$, onde cada posição contém um número entre 0 e 9 e cada linha da matriz representa um número de 5 dígitos. O algoritmo deve calcular a soma dos 5 números contidos na matriz colocando o resultado em um vetor Soma[1..6]. Escreva ao final o vetor Soma.

M00001300 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$, onde cada posição contém um número entre 0 e 9 e cada linha da matriz representa um número de 5 dígitos. O algoritmo deve encontrar a linha que contém o maior dos 5 números e escrever o número.

M00001400 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$, onde cada posição contém um número entre 0 e 9 e cada linha da matriz representa um número de 5 dígitos. O algoritmo deve ordenar os 5 números em ordem crescente, e escrever a matriz com os números ordenados.

M00001500 - Faça um algoritmo que leia dois valores D e DS, correspondendo a um dia do mês e um dia da semana (1-domingo, 2-segunda, ... 7-sábado) e preencha uma matriz Folhinha[1..6,1..7] com o calendário correspondente ao mês do dia digitado. A primeira coluna da matriz contém os domingos, a segunda coluna

contem as segundas e assim por diante. O algoritmo deve escrever a matriz gerada. As posições não utilizadas da matriz devem conter 0's.

M00001600 - Uma matriz de adjacências, para um mapa rodoviário, é composta de elementos 0's e 1's, sendo que $M[i,j] = 1$ se existe uma ligação rodoviária da cidade i para a cidade j e $M[i,j] = 0$ em caso contrário. Essa matriz será simétrica se para todo caminho i,j existir o caminho j,i (estradas de mão dupla). Uma característica dessa matriz é que na matriz M^2 , obtida pelo produto matricial de M por M , cada posição $M[i,j]$ contém o número de caminhos entre as cidades i e j que passam exatamente por 1 outra cidade. E M^3 contem a quantidade de caminhos que passam por 2 outras cidades. E assim por diante. Escreva um algoritmo que leia uma matriz de adjacências M para um conjunto de 5 cidades e, após, leia os valores $c1$ e $c2$ (ambos menores ou iguais a 5) representando duas cidades quaisquer do conjunto. O algoritmo deve descrever a quantidade de caminhos entre $c1$ e $c2$ que passam exatamente por 1 outra cidade.

M00001700 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e um valor N e multiplica cada valor da matriz M por N e coloca o resultado em um vetor $V[1..25]$. Escreva ao final o vetor V .

M00001800 - Faça um algoritmo que leia uma matriz $M[1..5,1..6]$ e divide todos os 6 elementos de cada linha pelo valor do menor elemento EM MÓDULO da linha. Escrever a matriz modificada.

M00001900 - Faça um algoritmo que leia uma matriz $M[1..5,1..5]$, possivelmente com elementos repetidos. Leia, a seguir, 5 valores e , para cada um, verifique se o valor ocorre ou não na matriz, escrevendo a posição (linha e coluna) em que foi encontrada a primeira ocorrência do mesmo e , caso ele não exista na matriz, a mensagem "Não tem".

M00002000 - Uma matriz é dita triangular superior se todos os elementos abaixo da diagonal principal são iguais a zero, e há pelo menos um elemento nulo acima da diagonal principal. Da mesma forma, uma matriz é dita triangular inferior se todos os elementos acima da diagonal principal são iguais a zero, e há pelo menos um elemento não nulo abaixo da diagonal principal. E uma matriz é dita diagonal se os elementos não nulos ocorrem somente na diagonal principal. Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e escreva: 0 - Se a matriz é uma matriz diagonal; 1 - Se é triangular superior; 2 - Se é triangular inferior; 3 - Se não é nenhuma das anteriores

M00002100 - Uma matriz é dita Matriz de Toeplitz se, em cada diagonal paralela à diagonal principal, todos os elementos são iguais. Assim, um exemplo de Matriz de Toeplitz é:

```
1 3 5 4 6
2 1 3 5 4
8 2 1 3 5
7 8 2 1 3
9 7 8 2 1
```

Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e verifique se é uma Matriz de Toeplitz, escrevendo: 1 - Se é uma matriz de Toeplitz; 0 - Se não é uma matriz de Toeplitz

M00002200 - Uma matriz é dita Circulante cada elemento é igual ao elemento imediatamente acima à esquerda, e se o primeiro elemento de cada coluna é igual ao último elemento da coluna anterior. Assim, um exemplo de Matriz Circulante é:

```
1 9 7 8 2
2 1 9 7 8
8 2 1 9 7
7 8 2 1 9
9 7 8 2 1
```

Faça um algoritmo que leia uma matriz $M[1..5,1..5]$ e verifique se é uma Matriz Circulante, escrevendo: 1 - Se é uma matriz Circulante; 0 - Se não é uma matriz Circulante

M00002300 - Faça um algoritmo que leia uma matriz $M[1..3,1..3]$ e para cada linha divida toda a linha pelo seu elemento da primeira coluna. Após isso, a partir da segunda linha, de cada linha subtraia toda a primeira linha elemento a elemento. ex. a matriz:

```
2 4 6
3 9 6
2 8 4
```

torna-se

```
1 2 3
1 3 2
1 4 2
```

e finalmente

```
1 2 3
0 1 -1
0 2 -1
```

Escreva a matriz alterada.

M00002400 - Uma matriz é dita de Hadamard se todos os seus elementos são 1 ou -1, e o produto escalar (somatório dos produtos de cada elemento de uma linha pelo elemento correspondente da outra linha) entre quaisquer duas linhas é igual a zero. Faça um algoritmo que leia uma matriz $M[1..4,1..4]$ e verifique se ela é de Hadamard escrevendo: 0 - Se não é de Hadamard; 1 - Se é de Hadamard

M00002500 - Faça um algoritmo que leia uma matriz $M[1..3,1..3]$ e rotacione a matriz 90 graus no sentido horário, escrevendo a matriz resultante.

M00004900 - Um quadrado mágico de ordem N (sendo N um número ímpar) é um arranjo de número de 1 a $N \times N$ em uma matriz quadrada de tal modo que a soma de cada linha, coluna e diagonal é a mesma. A matriz abaixo representa um quadrado mágico de ordem 5.

```
15  8  1 24 17
16 14  7  5 23
22 20 13  6  4
 3 21 19 12 10
 9  2 25 18 11
```

A regra para gerá-lo é relativamente fácil de observar: Comece com 1 no meio da primeira linha, então siga para cima e para a esquerda diagonalmente (quando sair do quadrado suponha que os lados superior e inferior estão unidos ou que os lados da direita e da esquerda estão unidos, conforme for o caso). Em cada quadrado que passar coloque o valor do quadrado anterior mais 1 (um). Quando a próxima casa estiver já preenchida, desça um quadrado e continue seguindo a diagonal até ter preenchido todos os quadrados. Escrever um algoritmo que lê um número N ímpar, menor ou igual a 99, e gere e escreva, para o número lido, o seu quadrado mágico.

M00005000 - Faça um algoritmo que leia duas matrizes $M[1..3,1..3]$ e $N[1..3,1..3]$ (primeiro a matriz M, e após a matriz N) e calcule o produto matricial $M \times N$ colocando o resultado em uma matriz $P[1..3,1..3]$. Escreva ao final a matriz P.

M00005100 - As pirâmides têm a base quadrada, sendo que a única forma de se atingir o topo é seguir em espiral pela borda. Escreva um algoritmo que leia um valor N e, a seguir, uma matriz quadrada $A[n,n]$ de ordem no máximo igual a 10, e

verifique se a matriz é equivalente a uma pirâmide inca, ou seja, se partindo-se do canto superior esquerdo da matriz, no sentido horário, em espiral, a posição seguinte na ordem é o inteiro consecutivo da posição anterior. O algoritmo deve escrever: 1 - Se a matriz forma uma pirâmide inca; 0 - Em caso contrário.

M00005200 - Escreva um algoritmo que leia duas matrizes numéricas $A[3 \times 4]$ e $B[3 \times 4]$ e gere e escreva uma matriz inteira $C[3 \times 4]$, tal que um elemento $C[i,j] = 1$ se os elementos nas mesmas posições das matrizes A e B forem iguais, e 0 em caso contrário. O algoritmo deve, ao final, escrever a matriz C.

M00005300 - Na Teoria de Sistemas, define-se como elemento minimax de uma matriz o menor elemento da linha em que se encontra o maior elemento da matriz. Escreva um algoritmo que leia uma matriz $A[5 \times 5]$ e determine o seu elemento minimax. O algoritmo deve, ao final, escrever o valor do elemento minimax, bem como a linha e coluna onde ocorreu.

M00005400 - (Problema específico da disciplina de Teoria de Grafos) Uma matriz de adjacências, para um mapa rodoviário, é composta de elementos 0's e 1's, sendo que $M[i,j] = 1$ se existe uma ligação rodoviária da cidade i para a cidade j e $M[i,j] = 0$ em caso contrário. Essa matriz será simétrica se para todo caminho i,j existir o caminho j,i (estradas de mão dupla). Matrizes de adjacências são uma das formas utilizadas para representação de grafos. Escreva um algoritmo que leia uma matriz de adjacências M para um conjunto de 5 cidades e, após, leia os valores c1 e c2 (ambos menores ou iguais a 5) representando duas cidades quaisquer do conjunto. O algoritmo deve descrever o menor caminho de c1 a c2, mostrando a cidade inicial, as cidades intermediárias e a cidade final.

24 Sequenciais

59 Condicionais

95 Iterativos

84 de Vetores

33 de Matrizes

total: 295 problemas