

Lista de Exercícios de Programação Estruturada

Q1) Implemente um programa que, infinitamente, receba, como parâmetro de entrada, um número n e retorne os n primeiros números primos existentes. Seu programa para quando n for menor ou igual a zero.

Q2) Implemente um programa que, infinitamente, receba, como parâmetro de entrada, um número n e retorne os n primeiros números primos existentes depois de n . Por exemplo, se $n = 2$, a resposta será os primos 3 e 5. É necessário salientar que n não precisa ser primo. Seu programa para quando n for menor ou igual a zero.

Q3) **[Problema 1103 do URI Online Judge]** Daniela é enfermeira em um grande hospital, e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores. Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar. No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso.

Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir.

Entrada: a entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros $H1$, $M1$, $H2$, $M2$, com $H1:M1$ representando a hora e minuto atuais, e $H2:M2$ representando a hora e minuto para os quais o alarme despertador foi programado ($0 \leq H1 \leq 23$, $0 \leq M1 \leq 59$, $0 \leq H2 \leq 23$, $0 \leq M2 \leq 59$). O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

Saída: para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Daniela tem para dormir.

Entrada	Saída
1 5 3 5	120
23 59 0 34	35
21 33 21 10	1417
0 0 0 0	

Q4) **[Problema 1028 do URI Online Judge]** Ricardo e Vicente são aficionados por figurinhas. Nas horas vagas, eles arrumam um jeito de jogar um “bafo” ou algum outro jogo que envolva tais figurinhas. Ambos também têm o hábito de trocarem as figuras repetidas com seus amigos e certo dia pensaram em uma brincadeira diferente. Chamaram todos os amigos e propuseram o seguinte: com as figurinhas em mãos, cada um tentava fazer uma troca com o amigo que estava mais perto seguindo a seguinte regra: cada um contava quantas figurinhas tinha. Em seguida, eles tinham que dividir as figurinhas de cada um em pilhas do mesmo tamanho, no maior tamanho que fosse possível para ambos. Então, cada um escolhia uma das pilhas de figurinhas do amigo para receber. Por exemplo, se Ricardo e Vicente fossem trocar as figurinhas e tivessem respectivamente 8 e 12 figuras, ambos dividiam todas as suas figuras em pilhas de 4 figuras (Ricardo teria 2 pilhas e Vicente teria 3 pilhas) e ambos escolhiam uma pilha do amigo para receber.

Entrada: a primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 3000$), indicando o número de casos de teste. Cada caso de teste contém 2 inteiros $F1$ ($1 \leq F1 \leq 1000$) e $F2$ ($1 \leq F2 \leq 1000$) indicando, respectivamente, a quantidade de figurinhas que Ricardo e Vicente têm para trocar.

Saída: para cada caso de teste de entrada haverá um valor na saída, representando o tamanho máximo da pilha de figurinhas que poderia ser trocada entre dois jogadores.

Entrada	Saída
3	4
8 12	9
9 27	37
259 111	

Q5) Implemente um programa que integre e derive polinômios. Cada polinômio é definido por um vetor que contém seus coeficientes. Por exemplo, o polinômio de grau dois $3x^2 + 2x + 12$ terá um vetor de coeficientes $v = \{12, 2, 3\}$. Sua integral será $I = \{0, 12, 1, 1\}$, equivalente ao polinômio $x^3 + x^2 + 12x$, e sua derivada será $D = \{2, 6\}$, equivalendo ao polinômio $6x + 2$. O programa deve, infinitamente: (a) receber o valor do maior grau g do polinômio, seguido de $(g + 1)$ coeficientes; (b) calcular qual é a integral e a derivada do polinômio informado; e (c) imprimir os novos polinômios na tela. Este programa para somente quando o grau g do polinômio for menor ou igual a zero.

Q6) Um número a é dito permutação de um número b se os dígitos de a formam uma permutação dos dígitos de b . Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455. Faça um programa que receba a e b e responda se a é permutação de b . Obs.: Considere que o dígito 0 (zero) não deve aparecer nos números.

Q7) Implemente um programa que, infinitamente, teste se um número é um palíndromo (Dica: se uma palavra pode ser lida, indiferentemente, da esquerda para a direita e vice-versa, ela é considerada um palíndromo). Você deve passar o número a a ser testado. O seu programa deverá imprimir as seguintes mensagens “VERDADEIRO” (caso o número seja um palíndromo) ou “FALSO” (caso o número não seja um palíndromo) na console. Seu programa para quando o número for negativo.

Q8) Implemente um programa que, infinitamente, receba, como parâmetro de entrada, um número n e retorne a representação binária de n . Por exemplo, se n é igual a 12, a resposta deste programa deve ser “1100”. Seu programa para quando n for menor que zero.

Q9) **[Problema 1318 do URI Online Judge]** Sua escola organizou uma grande festa para celebrar a brilhante vitória do seu time no prestigiado, e mundialmente famoso CCIP (Competição Colegial Internacional de Poesia). Todos na sua escola foram convidados para a noite, que incluía coquetel, jantar e uma sessão onde a poesia de seu time era lida para a audiência. O evento foi um sucesso – mais pessoas mostraram interesse em sua poesia do que você esperava – porém alguns de seus críticos disseram que tamanho público esteve presente graças à comida, e não graças a sua poesia.

Independente do motivo, no dia seguinte você descobriu o motivo pelo qual o salão esteve tão cheio: o diretor da escola lhe confidenciou que diversos dos bilhetes usados pelos visitantes eram falsos. O número real de bilhetes foram numerados sequencialmente de 1 a N ($N \leq 10000$). O diretor suspeita que algumas pessoas usaram o scanner e a impressora da Sala da Computação para produzir cópias dos bilhetes verdadeiros. O diretor lhe deu um pacote contendo todos os bilhetes coletados dos visitantes na entrada da festa, e lhe pediu para que determinasse quantos bilhetes no pacote continham “clones”, isto é, outro bilhete com o mesmo número da sequência.

A entrada contém dados de diversos casos de teste. Cada caso de teste contém duas linhas. A primeira linha contém dois inteiros N e M , que indicam, respectivamente, o número de bilhetes originais e o número de pessoas presentes na festa ($1 \leq N \leq 10000$ e $1 \leq M \leq 20000$). A segunda linha do caso de testes contém M inteiros T_i representando os números dos bilhetes contidos no pacote que o diretor lhe deu ($1 \leq T_i \leq N$). O final da entrada é indicado por $N = M = 0$. Para cada caso de teste seu programa deverá imprimir uma linha, contendo o número de bilhetes do pacote que contém outro bilhete com o mesmo número da sequência.

Entrada	Saída
5 5 3 3 1 2 4 6 10 6 1 3 6 6 4 2 3 1 2 0 0	1 4

Q10) **[Problema 1533 do URI Online Judge]** John Watson, mesmo após anos trabalhando ao lado de Sherlock Holmes, nunca conseguiu entender como ele consegue descobrir quem é o assassino com tanta facilidade. Em uma certa noite, porém, Sherlock bebeu mais do que devia e acabou contando o segredo a John. “Elementar, meu caro Watson”, disse Sherlock Holmes. “Nunca é o mais suspeito, mas sim o segundo mais suspeito”. Após descobrir o segredo, John decidiu resolver um crime por conta própria, só para testar se aquilo fazia sentido ou se era apenas conversa de bêbado. Dada uma lista com **N** inteiros, representando o quanto cada pessoa é suspeita, ajude John Watson a decidir quem é o assassino, de acordo com o método citado.

Haverá diversos casos de teste. Cada caso de teste inicia com um inteiro **N** ($2 \leq N \leq 1000$), representando o número de suspeitos. Em seguida haverá **N** inteiros distintos, onde o **i**-ésimo inteiro, para todo $1 \leq i \leq N$, representa o quão suspeita a **i**-ésima pessoa é, de acordo com a classificação dada por John Watson. Seja **V** o valor do **i**-ésimo inteiro, $1 \leq V \leq 10000$. O último caso de teste é indicado quando **N** = 0, o qual não deverá ser processado. Para cada caso de teste imprima uma linha, contendo um inteiro, representando o índice do assassino, de acordo com o método citado.

Entrada	Saída
3 3 5 2 5 1 15 3 5 2 0	1 4

Q11) Implemente uma função em C que receba uma string como parâmetro e substitua todas as suas letras por predecessoras do alfabeto, trocando maiúsculas por minúsculas. Por exemplo, a string “Amor” seria alterada para “zLNQ”. Esta função deve obedecer o seguinte protótipo: **void shift_troca_string (char *str)**. A letra ‘a’ deve ser substituída por ‘Z’ (e ‘A’ por ‘z’). Caracteres que não forem letras devem ser substituídos por ‘!’. Sabe-se que A = 65, Z = 90, a = 97 e z = 122.

Q12) Implemente uma função em C que receba uma string como parâmetro e retorne a string com todos os caracteres minúsculos substituídos pelo caracter ‘?’. Por exemplo, se for passada a string “740-Charitas-Leme”, a função deve retornar a string “740-Ch?r?t?s-L?m?”. A assinatura desta função deve ser **void codifica (char *str)**.

Q13) Considere que você receba um vetor de inteiros com números ímpares e pares e um número **n**, sendo **n** o tamanho do vetor. Implemente uma função que separe os valores inteiros de um vetor em ímpares e pares, obedecendo a ordem de entrada destes números no vetor: **void transf_i_p (int *vet, int n)**.

Q14) Escreva uma função que receba, como entrada, uma cadeia de caracteres **s** e um inteiro **n**, e, em seguida, retire o **"prefixo"** da cadeia **s** de tamanho **n** (isto é, retire os **n** primeiros caracteres). Se a cadeia não tiver pelo menos **n** caracteres, deve ser impressa a mensagem **"erro"**. Por exemplo, se **s** = **"abcdefghi"** e **n** = **3**, então a cadeia **"defghi"** deve ser impressa; com a mesma cadeia **s** e **n** = **17**, deve ser impresso **"erro"**. O protótipo desta função é o seguinte: **void retira_inicio_n (char *str, int n)**.

Q15) Escreva uma função que receba, como entrada, uma cadeia de caracteres **s** e um inteiro **n**, e, em seguida, retire o **"sufixo"** da cadeia **s** de tamanho **n** (isto é, retire os **n** últimos caracteres). Se a

cadeia não tiver pelo menos **n** caracteres, deve ser impressa a mensagem **"erro"**. Por exemplo, se **s = "abcdefghi"** e **n = 3**, então a cadeia **"abcdef"** deve ser impressa; com a mesma cadeia **s** e **n = 17**, deve ser impresso **"erro"**. O protótipo desta função é o seguinte: **void retira_fim_n (char *str, int n)**.

Q16) Escreva um programa que receba, como parâmetro de entrada, uma string, e verifique se esta string é uma "boa" senha. Entende-se por "boa" senha uma senha com as seguintes características de formação: (i) ela deve possuir, no mínimo, oito caracteres; (ii) ela deve ter, no mínimo, um caracter entre '0' a '9'; (iii) ela deve possuir, no mínimo, uma letra maiúscula; (iv) ela deve ter, no mínimo, uma letra minúscula; e (v) ela deve ter, pelo menos, um caracter do conjunto formato pelos demais caracteres do teclado.

Q17) Escreva um programa que receba duas strings, como parâmetros de entrada, e informe qual é a maior substring existente nas duas strings. Por exemplo, se as strings de entrada são ACCTGAACTCCCCC e ACCTAGGACCCCC, então a maior substring existente entre as duas strings será CCCCC.

Q18) Escreva uma função que dadas duas strings, retorne UM se a primeira contém a segunda, ignorando maiúsculas e minúsculas, e ZERO, caso contrário. O protótipo da função é o seguinte: **int cic (char *str1, char *str2)**.

Q19) Escreva uma função que dadas duas strings, retorne UM se as strings são iguais, ignorando maiúsculas e minúsculas, e ZERO, caso contrário. O protótipo da função é o seguinte: **int igual (char *str1, char *str2)**.

Q20) Dada uma matriz quadrada de dimensão 9, com valores de 1 a 9 em suas posições, escreva um programa que verifique se esta matriz é uma solução válida para o **Sudoku** (isto é, uma solução é válida no **Sudoku** se cada linha, cada coluna e cada bloco contém os números de 1 a 9 somente uma vez).

```
5 3 4 | 6 7 8 | 9 1 2
6 7 2 | 1 9 5 | 3 4 8
1 9 8 | 3 4 2 | 5 6 7
-----+-----+-----
8 5 9 | 7 6 1 | 4 2 3
4 2 6 | 8 5 3 | 7 9 1
7 1 3 | 9 2 4 | 8 5 6
-----+-----+-----
9 6 1 | 5 3 7 | 2 8 4
2 8 7 | 4 1 9 | 6 3 5
3 4 5 | 2 8 6 | 1 7 9
```