

Lista de exercícios

Programação e Desenvolvimento de Software 2

Revisão de C++

Projetos de programação retirados dos Capítulos 1 ao 5 de "C++ Absoluto", Walter J. Savitch, editora Pearson.

Cap. 1 (Projeto de Programação 6). Um empregado recebe R\$ 16,78 por horas regulares trabalhadas em uma semana. Se esse empregado fizer hora extra, deve receber essa mesma taxa multiplicada por 1,5. Do pagamento bruto do empregado, 6% são retidos pela Previdência Social, 14%, pelo Imposto de Renda Federal, 5%, por impostos estaduais, e R\$ 10 por semana, para o Sindicato. Se o empregado tiver três ou mais dependentes, um adicional de R\$ 35 é retido para cobrir o custo extra do seguro de saúde. Escreva um programa que leia o número de horas trabalhadas em uma semana e o número de dependentes como entrada e apresente como saída o pagamento bruto do empregado, o valor de cada imposto retido e o salário líquido por semana.

Cap. 2 (Projeto de Programação 2). Você acaba de adquirir um aparelho estereofônico que custa R\$ 1.000 por meio do seguinte plano de crediário: zero de entrada, juros de 18% ao ano (e, portanto, 1,5% ao mês) e prestações mensais de R\$ 50. A prestação mensal de R\$ 50 é utilizada para pagar os juros, e o restante é utilizado para pagar parte da dívida remanescente. Assim, no primeiro mês você paga 1,5% de R\$ 1.000 em juros. Isso dá R\$ 15. Os restantes R\$ 35 são deduzidos do seu débito, o que o deixa com um débito de R\$ 965,00. No mês seguinte você paga um juro de 1,5% sobre R\$ 965,00, que dá R\$ 14,48. Assim, você pode deduzir R\$ 35,52 (que é R\$ 50 – R\$ 14,48) da soma que deve.

Escreva um programa que lhe diga quantos meses você levará para pagar o que deve, assim como a soma total paga em juros. Utilize um loop para calcular a soma paga em juros e o tamanho do débito a cada mês. (Seu programa final não precisa fornecer a quantia paga mensalmente a título de juros, mas você pode querer escrever uma versão preliminar do programa que apresente esses valores.) Utilize uma variável para contar o número de iterações do loop e, portanto, o número de meses até que o débito seja zero. Você pode querer utilizar outras variáveis também. O último pagamento pode ser inferior a R\$ 50 se o débito for menor, mas não se esqueça dos juros. Se você deve R\$ 50, então sua prestação mensal de R\$ 50 não saldará seu débito, embora vá chegar perto disso. Os juros de um mês sobre R\$ 50 são de apenas 75 centavos.

Cap.3 (Projeto de Programação 7). Quando está frio, os meteorologistas transmitem um índice chamado fator de frio do vento, que leva em consideração a velocidade do vento e a temperatura. O índice fornece uma medida do efeito resfriador do vento em uma dada temperatura do ar. Esse índice pode ser aproximado pela seguinte fórmula:

$$W = 13.12 + 0.6215*t - 11.37*v^{0.16} + 0.3965*t*v^{0.016}$$

em que

v = velocidade do vento em m/s

t = temperatura em graus Celsius: $t \leq 10$

W = índice de frio do vento (em graus Celsius)

Escreva uma função que forneça o índice de frio do vento. Seu código deve assegurar que a restrição a respeito da temperatura não seja violada. Verifique alguns boletins meteorológicos em edições anteriores de jornais em sua biblioteca e compare o índice de frio do vento que você calculou com o resultado divulgado no jornal.

Cap.4 (Projeto de Programação 7). (a) Escreva um programa que leia o peso em libras (1 libra = 453,59 gramas) e onças (1 onça = 28,34 gramas) e apresente como saída o equivalente em quilogramas e gramas. Use pelo menos três funções: uma para entrada, uma ou mais para o cálculo e uma para a saída. Inclua um loop que permita ao usuário repetir esse cálculo para novos dados de entrada até o usuário dizer que deseja encerrar o programa. Existem 2.2046 libras em um quilograma, 1.000 gramas em um quilograma e 16 onças em uma libra. (b) Escreva um programa como o do exercício anterior que converta quilogramas e gramas em libras e onças. Utilize funções para as subtarefas.

Cap 5 (Projeto de Programação 3).

O desvio-padrão de uma lista de números é a medida de quanto os números se desviam da média. Se o desvio-padrão é pequeno, os números estão aglomerados junto à média. Se o desvio-padrão é grande, os números estão dispersos em relação à média. Defina uma função que tome um vetor parcialmente preenchido de números e seu argumento e retorne o desvio-padrão dos números no vetor parcialmente preenchido. Como um vetor parcialmente preenchido requer dois argumentos, a função, na realidade, terá dois parâmetros formais: um parâmetro vetorial e um parâmetro formal de tipo int que fornece o número de posições vetoriais utilizado. Os números no vetor serão de tipo double. Insira sua função em um programa-teste adequado.

Cap 5 (Projeto de Programação 9). O matemático John Horton Conway inventou o "Jogo da Vida". Embora não seja um "jogo" no sentido tradicional, ele apresenta um comportamento interessante, especificado com poucas regras. Esse projeto pede que você escreva um programa que lhe permita especificar uma configuração inicial (lida de um arquivo). O programa segue as regras da Vida (listadas brevemente) para mostrar o comportamento contínuo da configuração. VIDA é um organismo que vive em um mundo distinto, bidimensional. Embora esse mundo seja, na realidade, ilimitado, não temos toda essa liberdade e, assim, restringimos o vetor a 80 caracteres de largura e 22 de altura. Se você tem acesso a uma tela maior, use-a! Esse mundo é um vetor em que cada célula é capaz de abrigar uma célula da VIDA. As gerações marcam a passagem do tempo. Cada geração traz nascimentos e mortes para a comunidade da VIDA. Os nascimentos e mortes seguem o conjunto de regras:

1. Cada célula possui oito células vizinhas. As vizinhas de uma célula são as células diretamente acima, abaixo, à direita, à esquerda, diagonalmente acima à direita ou à esquerda e diagonalmente abaixo à direita ou à esquerda.
2. Se uma célula ocupada não possui vizinhas ou possui apenas uma, morre de solidão. Se uma célula ocupada possui mais de três vizinhas, morre de superpopulação.
3. Se uma célula vazia possui exatamente três células vizinhas ocupadas, há o nascimento de uma nova célula para substituir a célula vazia.
4. Nascimentos e mortes são instantâneos e ocorrem com as mudanças de geração. Uma célula que morre por qualquer razão pode ajudar a provocar o nascimento, mas uma célula recém-nascida não pode ressuscitar uma célula que está morrendo, nem a morte de uma célula impede a morte de outra, digamos, por meio da redução da população local.

Exemplos em: <https://www.youtube.com/watch?v=23MBR2pZoDQ>

Observações: algumas configurações crescem a partir de configurações iniciais bem pequenas. Outras se des-locam pela região. Recomenda-se que, para a saída de texto, você utilize um vetor retangular de char com 80 colunas e 22 linhas para armazenar as sucessivas gerações mundiais de VIDA. Utilize um * para indicar uma célula viva e um espaço em branco para indicar uma célula vazia (ou morta). Se você possui uma tela com mais linhas do que isso, não hesite em utilizar a tela toda. Sugestões: procure configurações estáveis. Ou seja, procure por comunidades que repitam os padrões continuamente. O número de configurações na repetição é chamado de período. Há configurações que são fixas, ou seja, que permanecem sem mudança. Um plano possível é encontrar essas configurações. Dicas: defina uma função void chamada geração que tome o vetor que chamamos mundo, um vetor de tipo char de 80 colunas por 22 linhas, que contenha a

configuração inicial. A função percorre o vetor e modifica as células, assinalando as células com nascimentos e mortes de acordo com as regras listadas anteriormente. Isso envolve examinar uma célula de cada vez e ou matar a célula, ou deixá-la viver ou, se a célula estiver vazia, decidir se uma célula deve nascer. Deve haver uma função mostra que aceite o vetor mundo e exiba o vetor na tela. É preciso haver alguma espécie de intervalo de tempo entre as chamadas a geração e a mostra. Para fazer isso, seu programa deve gerar e mostrar a próxima geração quando se aperta a tecla Return. Você é livre para automatizar isso, mas a automação não é necessária para o programa.