UFV - Universidade Federal de Viçosa
DPI - Departamento de Informática
Prof. Alcione

INF 110 - Programação I
Trabalho 1
Prazo: 09/10/2020

O trabalho consiste em projetar e implementar programas em C ou C++ para resolver os 5 problemas descritos adiante. Não há necessidade de entregar os códigos impressos ou por e-mail, nem entregar algum relatório. Os códigos devem ser submetidos para correção no sistema Boca, e para cada problema podem ser feitas tantas submissões quantas forem necessárias até obter uma correta. No caso de mais de uma submissão correta para um problema, será considerada na avaliação principalmente a última.

ATENÇÃO!

Este trabalho é **individual**.

Duas importantes etapas na resolução de cada problema são o planejamento da lógica do algoritmo e a implementação dessa lógica em uma linguagem de programação. Antes de buscar alguma ajuda na resolução do trabalho, considere os seguintes pontos:

1. Se alguém te contar a lógica que desenvolveu antes de você tentar desenvolver uma, perderá a oportunidade de exercitar esta etapa.

2. Se alguém te mostrar ou passar uma solução em código C ou C++ antes de você tentar desenvolver ou encontrar os erros na sua implementação, perderá a oportunidade de exercitar e desenvolver suas habilidades de programação.

Portanto, tente desenvolver por si mesmo a lógica do algoritmo de solução. Se não conseguir resolver uma questão completamente, busque ajuda do professor ou de algum colega, mas não olhe um código pronto.

E quando terminar um código jamais passe esse código para um colega. Ajude-o explicando a lógica da solução, ou encontrando erros na implementação, jamais entregue seu código a outros alunos. Trabalhos com códigos idênticos ou muito parecidos receberão nota **ZERO**.

Comentários e indentação

Códigos que não estejam devidamente indentados podem receber desconto na nota. Além disso, devem existir comentários explicativos nas partes mais complexas da solução.

Data limite

A data limite de entrega será quinta-feira, dia 22 de abril. Ainda falta bastante tempo, mas comece a fazê-lo assim que receber este enunciado para evitar problemas de última hora.

Link para o envio dos programas no sistema Boca

http://boca.dpi.ufv.br/trabalhos
Name: seu número de matrícula
Password: seu número de matrícula
ATENÇÃO! **Troque sua password!** (link Options no menu do Boca)

Problema A. Estações do ano

Arquivo-fonte: `estacoes.c` ou `estacoes.cpp`

Pouca gente percebeu, mas há algumas semanas começou oficialmente o Outono. Mais precisamente, no dia 20 de março. A data exata de início de cada estação pode variar ligeiramente de ano pra ano, e é definida pelos solstícios e equinócios. Em 2019, as datas de início das estações são as seguintes:

Outono	Inverno	Primavera	Verão
20 de março	21 de junho	23 de setembro	22 de dezembro

Apesar de haver essas datas oficiais, as quatro estações existem de fato apenas na região sul, nos estados de São Paulo e Mato Grosso do Sul e em regiões de serra no Rio de Janeiro e Minas Gerais. Na Amazônia, por exemplo, não há distinção significativa de temperatura nem de quantidade de chuvas ao longo do ano. No restante do país existem apenas duas estações, a seca e a chuvosa. Exceto em Viçosa, onde se pode vivenciar as quatro estações num único dia...

Sua tarefa é fazer um programa que lê uma data (dia e mês) e escreve a estação correta do ano, de acordo com a tabela acima.

Entrada

A entrada é composta por uma linha contendo dois inteiros, D e M, que representam o dia e o mês. Restrição: D e M representam uma data válida.

Saída

Seu programa deve gerar duas linhas na saída: a primeira contém a data por extenso, no formato "D `de` <*mês*>", sendo D o dia fornecido na entrada e <*mês*> o nome do mês M, com inicial minúscula; a segunda linha deve conter uma das palavras, "`Primavera`", "`Verao`", "`Outono`" ou "`Inverno`", dependendo da estação da data da entrada.

Não use cedilhas nem acentos na saída. Veja os exemplos.

Exemplos

Entrada	Saída
`4 4`	`4 de abril` `Outono`

Entrada	Saída
`25 12`	`25 de dezembro` `Verao`

Entrada	Saída
`20 3`	`20 de marco` `Outono`

Entrada	Saída
`19 3`	`19 de marco` `Verao`

Entrada	Saída
`1 10`	`1 de outubro` `Primavera`

Problema B. Esqui em Krasnoyarsk

Arquivo-fonte: `esqui.c` ou `esqui.cpp`

No mês passado ocorreu a XXIX Universíada de Inverno, em Krasnoyarsk, na Rússia. A Universíada é um evento multiesportivo internacional, organizado para atletas universitários. A versão de inverno ocorre a cada 2 anos e conta com esportes desconhecidos para a maioria dos brasileiros, já que são esportes praticados na neve ou sobre o gelo. Alguns são até conhecidos, mas não as regras de pontuação. Nesta questão você deve fazer um programa para calcular o *score* de um competidor na modalidade "Salto de esqui".

Nessa modalidade o atleta desce de esqui uma rampa e salta da rampa tentando aterrissar o mais distante possível (sem fazer acrobacias como no *free style*). O *score* leva em conta a distância e o estilo do salto.

Para a distância, existe uma marca (linha) alvo, chamada *K point*. Se o atleta aterrissa na marca, ganha 60 pontos. Se não aterrissa na marca ganha ou perde 1.8 pontos por cada metro além ou aquém da marca. Assim, numa competição K-120, em que a marca está a 120 metros, um atleta que aterrissa na marca dos 120 metros ganha 60 pontos. Outro que aterrissa a 122 metros ganha 63.6 pontos (60 + 3.6 pontos pelos 2 metros depois da marca), e um que aterrissa a 117 metros ganha 54.6 pontos (60 – 5.4 pontos pelos 3 metros antes da marca).

Para o estilo, existem 5 juízes que avaliam a posição do esqui durante o salto, o equilíbrio e postura do corpo do atleta e a aterrissagem. Cada um dá uma nota de valor no máximo 20. A nota do estilo é a soma das notas atribuídas pelo juízes, desconsiderando-se a menor e a maior das notas. Assim, um atleta que recebeu dos juízes as notas 19 19 18 18.5 e 19 terá nota de estilo 56.5, pois serão descartadas a nota 18 (menor) e uma das notas 19 (maior).

O *score* do atleta é a soma da nota da distância com a nota de estilo. Você deve fazer um programa que lê os dados do salto – distância e notas dos juízes, calcula e escreve o score do atleta, considerando uma competição K-120.

Entrada

A entrada contém os dados do salto de um atleta, descritos em duas linhas: a primeira contém um valor real D, que indica a distância do salto, em metros; a segunda contém cinco valores reais, J_1, J_2, J_3, J_4, J_5, que são as notas dos juízes. Restrições: $0 \leq D \leq 250$ e $0 \leq J_1, J_2, J_3, J_4, J_5 \leq 20$.

Saída

Seu programa deve gerar apenas uma linha de saída, contendo um valor real de uma casa decimal, representando o *score* do atleta, calculado conforme as regras descritas no enunciado.

Exemplos

Entrada	Saída
120 17 17 18 17 17	111.0

Entrada	Saída
135 18 18.5 18.5 19 19	143.0

Entrada	Saída
111 18 17 17 16.5 17	94.8

Problema C. Máquina automática

Arquivo-fonte: `maquina.c` ou `maquina.cpp`

O DCE quer instalar uma máquina automática de venda de chocolates, salgadinhos, refrigerantes, etc. Você foi convidado a programar o software que contabiliza o valor inserido pelo comprador e diz se ele é suficiente para comprar o produto escolhido.

Nesta primeira versão os valores estarão todos em centavos. Por exemplo, o valor de um produto que custa R$2,50 será representado por 250. E uma moeda de R$1,00 terá seu valor representado por 100.

Entrada

A primeira linha da entrada contém um valor inteiro P, que é o preço do produto escolhido. Em seguida virá uma linha contendo uma lista de valores V, cada um representando o valor de uma moeda inserida na máquina. A lista termina quando aparecer um valor 0, indicando que o comprador apertou o botão "liberar produto e troco". Restrição: $1 \leq P \leq 5000$, $V \in \{1, 5, 10, 25, 50, 100\}$.

Saída

Seu programa deve gerar uma linha na saída, que pode ser de dois tipos:

- se o comprador não inseriu dinheiro suficiente para comprar o produto, escreva "`Saldo insuficiente.`".
- caso contrário escreva "`Troco de` X `centavos.`", sendo X o valor apropriado.

Exemplos

Entrada	Saída
100 50 25 10 10 10 0	Troco de 5 centavos.

Entrada	Saída
105 50 25 10 10 10 0	Troco de 0 centavos.

Entrada	Saída
110 50 25 10 10 10 0	Saldo insuficiente.

Entrada	Saída
263 50 50 100 100 0	Troco de 37 centavos.

Entrada	Saída
199 25 5 10 100 100 0	Troco de 41 centavos.

Problema D. Máquina plus

Arquivo-fonte: `maquinap.c` ou `maquinap.cpp`

Agora que seu software para cálculo do valor pago pelo comprador foi testado e funciona corretamente, ele deve ser modificado para também aceitar cédulas e não apenas moedas.

A cada valor inserido pelo comprador, o software receberá uma letra e um valor inteiro. A letra será '`C`' ou '`M`', identificando se foi inserida uma cédula ou uma moeda. Caso seja uma cédula, o valor indica o valor da cédula, em reais. Caso seja uma moeda, indica o valor da moeda, em centavos. Assim, `C 1` representa a inserção de uma cédula de 1 real, enquanto `M 1` representa a inserção de uma moeda de 1 centavo. Note que a moeda de 1 real é representada por `M 100`.

O objetivo é o mesmo da questão anterior: contabilizar o valor total inserido e informar o troco, se houver. Além da possibilidade de inserir cédulas, o software deve considerar os produtos e valores pré-definidos da lista a seguir.

Número	Descrição do produto	Preço
1	Batata Shuffles	R$ 4,30
2	Suco C Mais+	R$ 2,70
3	Guaraná QWERTY	R$ 1,43

Entrada

A primeira linha da entrada contém um valor inteiro P, que é o número do produto escolhido. Em seguida virá uma sequência de linhas, cada uma contendo um caractere T e um valor V, representando respectivamente se foi inserida uma cédula ou moeda e o valor dela. A lista termina quando aparecer um valor 0, seja pra cédula ou moeda, indicando que o comprador apertou o botão "liberar produto e troco". Restrições: $1 \leq P \leq 10$, $T \in \{$'`C`', '`M`' $\}$, V um valor válido de cédula ou moeda brasileira.

Saída

Seu programa deve gerar uma linha na saída, que pode ser de três tipos:

- se o comprador escolheu um produto diferente de 1, 2 e 3, escreva a mensagem "`Produto inexistente.`"
- se o comprador não inseriu dinheiro suficiente para comprar o produto, escreva "`Saldo insuficiente.`".
- caso contrário escreva "`Troco de` X `centavos.`", sendo X o valor apropriado.

Exemplos

Entrada	Saída
1 C 5 C 0	Troco de 70 centavos.

Entrada	Saída
2 C 2 M 50 M 25 M 0	Troco de 5 centavos.

Entrada	Saída
4 C 100 C 20 C 0	Produto inexistente.

Entrada	Saída
3 C 10 C 0	Troco de 857 centavos.

Problema E. Máquina plus plus

Arquivo-fonte: `maquinapp.c` ou `maquinapp.cpp`

Como última funcionalidade do software antes de instalar a máquina no DCE, você deve implementar a parte que "devolve" o troco. Nesse caso, instruções que serão enviadas para o mecanismo da máquina liberar cédulas e moedas.

Da mesma forma que no problema anterior, o consumidor escolhe um produto e insere cédulas e moedas. Após verificar se o produto existe e se foi inserido um total suficiente, deve-se imprimir as instruções de troco, que são uma sequência de linhas contendo um caractere '`C`' ou '`M`', identificando se deve ser devolvida uma cédula ou moeda, e o valor (da cédula ou moeda).

Os valores válidos são:

```
C 100, C 50, C 20, C 10, C 5, C 2,
M 100, M 50, M 25, M 10, M 5, M 1.
```

Note que a máquina não devolve cédulas de R$ 1, pois já saíram de circulação, mas devolve moedas de R$ 1 (representadas por M 100).

Para minimizar o número de cédulas e moedas devolvidas a máquina devolve os valores na ordem acima (do maior ao menor valor). Ou seja, um troco de 30 centavos será devolvido como `M 25 M 5`, e não como `M 10 M 10 M 10` e nem como `M 5 M 25`.

Entrada

A entrada segue exatamente o mesmo formato da questão anterior.

Saída

A saída gerada pelo seu programa pode ser de três tipos:

- se o comprador escolheu um produto diferente de 1, 2 e 3, escreva a mensagem "`Produto inexistente.`"
- se o comprador não inseriu dinheiro suficiente para comprar o produto, escreva "`Saldo insuficiente.`".
- caso contrário, o programa deve simular a devolução do troco imprimindo o valor das cédulas e moedas devolvidas, uma em cada linha, na ordem especificada acima.

Exemplos

Entrada	Saída
1 C 5 C 0	M 50 M 10 M 10

Entrada	Saída
2 C 2 M 50 M 25 M 0	M 5

Entrada	Saída
4 C 100 C 20 C 0	Produto inexistente.

Entrada	Saída
3 C 50 C 0	C 20 C 20 C 5 C 2 M 100 M 50 M 5 M 1 M 1