
CONCURSO P7 – 1ª Festa da Programação

Regras da Primeira Festa da Programação

Deverão ter em conta as seguintes regras, as quais serão tidas em conta na validação dos programas:

- Em cada função, só poderão utilizar no máximo um ciclo (while ou for).
- Em cada função, só poderão utilizar no máximo uma instrução if – else, simples ou em cascata, ou uma instrução if sem else.
- Podem usar um ciclo (while ou for) e uma instrução if-else na mesma função.
- Só podem utilizar um e apenas um return em cada função.
- Todos os programas têm obrigatoriamente de ter uma função de teste.
- Todas as funções devolvem um valor inteiro ou double, exceto a função de teste.
- Programas aceites que não cumpram estas regras serão penalizados.

Programa A – Cereais



Portugal importa cereais de diversos pontos do mundo e tem de efetuar o seu transporte para o nosso País. Para diminuir custos, o ministério da agricultura determinou que, consoante as distâncias, o tipo de transporte será:

- Por via terrestre se a distância for inferior ou igual a 1000 km.
- Por via aérea se a distância for maior que 1000 km e menor ou igual a 3000 km.
- Por via marítima se a distância for maior que 3000 km.

Para calcular a distância usamos as coordenadas da origem e as coordenadas da cidade de Lisboa, que são Latitude 38.72 norte e Longitude 9.14 oeste.

Nota: as Latitudes norte são dadas por número positivos e as latitudes sul são dada por números negativos. Analogamente, as longitudes oeste são dadas por número positivos e as longitudes leste são dadas por números negativos.

Utilizaremos a seguinte fórmula, aproximada:

$$Distância_{km} = \sqrt{(LatLisboa - LatOrigem)^2 + (LongLisboa - LongOrigem)^2} \times 100$$

Sugestão: para calcular a raiz quadrada, use a função `sqrt`, fazendo também `#include <math.h>`.

A fórmula é aproximada, porque não considera a curvatura da terra...

No caso de via terrestre, a distância calculada deve depois ser corrigida para cima com um fator de 12%, para considerar “as curvas”. No caso de via aérea, deve ser corrigida para baixo por um fator de 25%, por o caminho ser otimizado. No caso de via marítima, deve ser corrigida para cima por um fator de 18%, para compensar a necessidade de contornar os países e evitar tempestades. Queremos uma função de teste que repetidamente receba dois números `double` representando a latitude e longitude do local de origem e que, para cada par de coordenadas lidas, escreva numa linha a distância aproximada calculada, a expressão “VT” ou “VA” ou “VM”, consoante dever usar-se via terrestre, via aérea ou via marítima, e a distância corrigida. Cada valor é separado do seguinte por um espaço. Os números `double` devem ser escritos com o formato `%.3f`. **Submeta no Problema A.**

Programa B – Tesla



Comprei um Tesla, modelo S, que é um carro fantástico. Tratando-se de um carro elétrico, tenciono carregá-lo em casa, porque fica muito mais barato. No entanto, ainda não sei se devo optar pela tarifa simples ou pela tarifa bi-horária.

Com a tarifa simples pagarei 0.23770€ por kWh, a qualquer hora do dia; com a tarifa bi-horária pagarei 0.26120€ por kWh, quando carregar de dia, entre as 7:00 e as 20:00, e pagarei 0.20620€ por kWh quando carregar de noite, entre as 20:00 e as 7:00 do dia seguinte.

Além disso, infelizmente há uma taxa de 25% aplicada ao valor dos kWh consumidos e tenho de pagar mensalmente a taxa de potência, que custa 15.903 euros para a tarifa simples e 18.551 para a tarifa bi-horária. Sobre o valor total, incide o IVA, a 23%. 😞

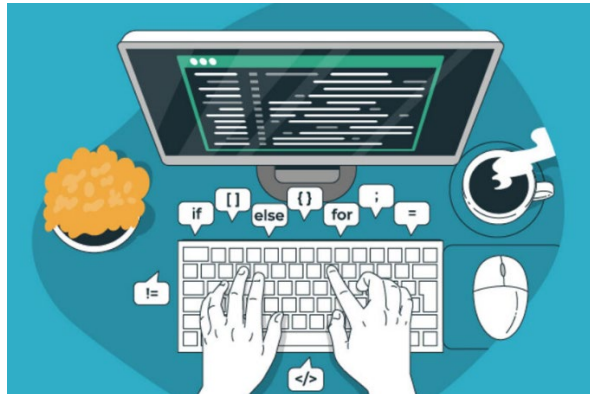
Preciso ponderar bem.

Para me ajudar nisso, vou escrever um programa para estimar a minha despesa mensal em eletricidade, admitindo que opto pela tarifa bi-horária, que parece mais vantajosa.

Quero uma função **bi_hourly**, que, dados os meus consumos mensais de eletricidade entre as 7:00 e as 20:00 e entre as 20:00 e as 7:00, calcule a minha despesa total (incluído IVA), baseada na tarifa bi-horária. Para verificar se esta é mais vantajosa, quero também uma função **simple**, análoga à outra (com os mesmos argumentos) que calcule quanto pagaria com a tarifa simples. Complementarmente, preciso de uma função de teste que repetidamente aceite do terminal dois números **double**, representando os meus consumos mensais de eletricidade durante o dia e durante a noite, até ao fim dos dados, e mostre para cada par de valores, os resultados calculados para esses valores pelas duas funções **simple** e **bi-hourly**.

Os dois resultados calculados devem ser escritos numa linha, separados por um espaço, usando formato **%.2f**. **Submeta no Problema B.**

Programa C –Programação Imperial



A minha cadeira preferida é Programação Imperial. Nesta cadeira, temos concursos de programação todas as semanas. O número de problemas dos concursos é variável, mas cada problema “aceite” vale 3 pontos. Por precaução, guardo o registo do número de problemas que vou conseguindo resolver, para saber quantos pontos já tenho.

Ora, o professor mudou as regras a meio do jogo, e disse que afinal quem em cada concurso resolvesse pelo menos metade dos problemas teria um ponto suplementar, de bónus.

Tenho de refazer as minhas contas. Ou, melhor, escreverei um programa para calcular por mim. Esse programa basear-se-á numa função **points**, com o seguinte cabeçalho:

int points(const int *contests, int n). O array **contests** contém os meus registos, com o número de problemas de cada concurso e o número de problemas que resolvi. Se já tiver havido **X** concursos, o array terá **2*X** números. É este o valor que virá no segundo argumento, **n**. No array **contests**, os primeiros **X** números representam o número de problemas em cada um dos **X** concursos; os restantes **X** números representam o número de problemas que resolvi em cada concurso (pela mesma ordem, bem entendido). Sugestão: para ler todos os números da consola para um array, recorra à função **ints_get**, estudada nas aulas. O número de números a ler não é fixo, mas será um número par, claro. Neste momento, os meus registos têm os seguintes números (lidos numa única linha de entrada de dados, aqui separados para melhor entendimento do que é solicitado):

5 4 2 6 6 3 5

5 2 0 2 3 1 3

Terei pontos de bónus no primeiro concurso, no segundo, no quinto e no sétimo. Portanto, o meu número de pontos total será **(5+2+0+2+3+1+3)*3+4**.

Para ter um programa completo, preciso ainda da função **main**, e dos **#include**, como de costume. **Submeta no problema C.**