

**Questão 1 (\*\*): Código Morse**

[35 pontos]

O *código morse* é um sistema de representação de letras, números e sinais de pontuação através de um sinal codificado. Foi desenvolvido por Samuel Morse em 1835, criador do telégrafo elétrico. A tabela seguinte apresenta a codificação para os dígitos numéricos.

Dígito	Código	Dígito	Código
1	.----	6	-....
2	..---	7	--...
3	...--	8	---..
4	....-	9	-----
5	.....	10	-----

Seu objetivo nessa questão consiste em escrever um programa C capaz de gerar e exibir na tela o código morse equivalente a um número inteiro e positivo no intervalo  $[0, 99]$  digitado pelo usuário em tempo de execução. Desconsidere a possibilidade de o usuário cometer erro ao informar o número. A entrada e a saída na tela devem seguir o exemplo apresentado abaixo. Chame o programa de `questao1.c`

**Exemplo de execução:**

Numero: 23

Morse: ..--- ...--

**Questão 2 (\*): Qual é o ângulo?**

[35 pontos]

Normalmente o aluno está acostumado a resolver problemas de Trigonometria onde é conhecido um ângulo e deseja-se saber o valor do seno, co-seno ou tangente. Porém, nada impede que se formule o problema inverso, ou seja, dado o valor do seno, por exemplo, seja calculado o respectivo ângulo.

Seu objetivo nessa questão é escrever um programa C para calcular e exibir na tela o ângulo em *graus* correspondente a um valor de *seno* fornecido pelo usuário em tempo de execução.

(continua no verso)

A biblioteca `<math.h>` possui uma função chamada `asin()` que recebe como argumento o valor do seno e que retorna o correspondente ângulo em *radianos*. Lembre-se que  $\pi$  radianos =  $180^\circ$ . Chame o programa de `questao2.c`

A entrada fornecida pelo usuário deve ser um número de ponto flutuante contido no intervalo  $[-1,1]$ . Caso o usuário informe um valor fora do intervalo, o programa deve exibir na tela uma mensagem de erro. O valor do ângulo em graus deve ser exibido com 4 dígitos decimais conforme exemplo que se segue.

#### Exemplo de execução:

Seno: 0.346738

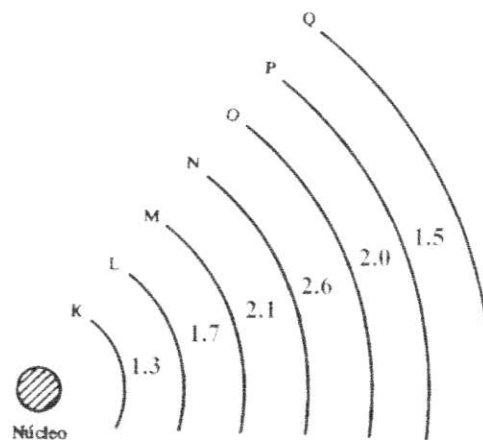
Angulo: 20.2879 graus

---

### Questão 3 (\*\*): Níveis Orbitais

[30 pontos]

Em um átomo, os elétrons orbitam em diferentes níveis de energia ao redor do núcleo. Atualmente são conhecidos 7 níveis de energia, onde K é o nível mais próximo do núcleo e Q é o mais distante, conforme apresentado abaixo.



Um elétron pode “pular” de um nível mais interno para outro mais externo se receber energia suficiente. Hipoteticamente, imagine que a quantidade de energia necessária para que um elétron mude para o nível externo seguinte dependa de um fator numérico conforme representado na figura anterior. Por exemplo, para passar do nível K para L é necessário 1.3 unidade de energia.

Seu objetivo nessa questão é escrever um programa C que solicite ao usuário a quantidade de energia recebida por um elétron inicialmente na camada K e que calcule e exiba na tela em qual camada ficará orbitando. A quantidade de energia deve ser um número de ponto flutuante. Desconsidere qualquer possibilidade de o usuário digitar um valor inválido na entrada de dados. Chame o programa de `questao3.c`

#### Exemplo de execução:

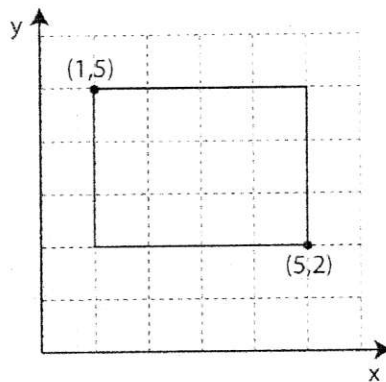
Energia: 8.2

Camada: 0

**Questão 1 (\*): Área do Retângulo**

[40 pontos]

Um retângulo pode ser definido no plano cartesiano a partir de dois de seus pontos extremos localizados em diagonais opostas, conforme ilustrado na figura abaixo:



Seu objetivo nessa questão é escrever um programa C que solicite ao usuário as coordenadas de dois pontos quaisquer do plano cartesiano e que calcule e exiba na tela a área do retângulo. Os pontos informados pelo usuário devem ser considerados como os extremos de diagonais opostas. Considere todos os valores fornecidos como tipos inteiros. Desconsidere a possibilidade de erro na entrada de dados. Chame o programa de `questao1.c`

**Exemplo de execução:**

Ponto 1: 1 5

Ponto 2: 5 2

Area: 12

**Questão 2 (\*\*\*): As Horas Pelo Mundo**

[30 pontos]

Na maioria dos países do mundo, inclusive no Brasil, é adotado o padrão de horário que divide o dia em 24 horas. Por exemplo, essa prova inicia-se às 18h50min. Entretanto, algumas nações como os Estados Unidos e o Reino Unido ainda empregam o padrão de 12 horas. Nesses países, diria-se que essa prova tem início às 06:50 PM.

(continua no verso)

No padrão 12 horas, as horas anteriores ao meio-dia são denominadas *ante meridiem* (AM) e as horas após o meio-dia são denominadas *post meridiem* (PM). Por convenção, 12h59min equivale a 12:59 AM e 13h equivale a 01:00 PM. De forma análoga, 0h59min equivale a 12:59 PM e 1h equivale a 01:00 AM.

Seu objetivo nessa questão consiste em escrever um programa C que solicite ao usuário a digitação de um horário no padrão 12 horas e que converta e exiba o horário para o padrão 24 horas. Desconsidere a possibilidade de erro na entrada de dados. Chame o programa de `questao2.c`

#### Exemplo de execução:

Padrao 12: 09:10 PM

Padrao 24: 21h10min

---

#### Questão 3 (\*\*): Algarismos Romanos

[30 pontos]

Os algarismos romanos utilizam caracteres do alfabeto latino para a representação de valores numéricos, conforme apresentado na tabela que se segue.

Romanos	Arábicos	Romanos	Arábicos
I	1	VI	6
II	2	VII	7
III	3	VIII	8
IV	4	IX	9
V	5	X	10

Seu objetivo nessa questão é escrever um programa C que solicite ao usuário a digitação de um valor em algarismos romanos (entre *I* e *X*) e que converta e exiba na tela o valor em algarismos arábicos. Observe que o usuário pode digitar os algarismos romanos em letras maiúsculas ou minúsculas. Desconsidere a possibilidade de erro na entrada de dados. Chame o programa de `questao3.c`

#### Exemplo de execução:

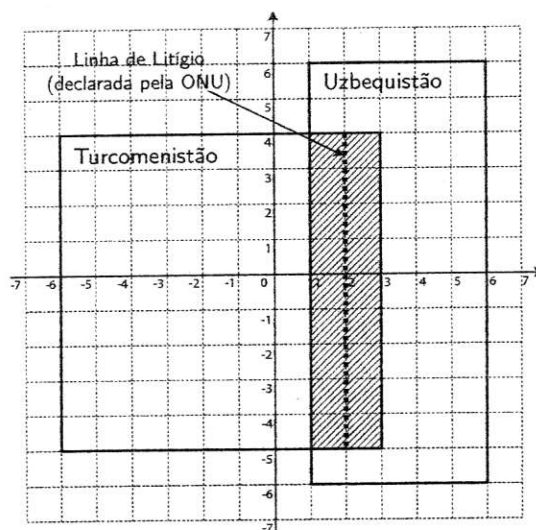
Romanos: vii

Arabicos: 7

### Questão 1 (\*\*): Guerra do Petróleo

[30 pontos]

O Turcomenistão e o Uzbequistão declararam guerra! O motivo foi a descoberta de uma grande bacia petrolífera na região fronteira entre os dois países asiáticos. O mapa que se segue identifica em tom de cinza a área litigiosa, rica em petróleo e contestada pelas duas nações.



Depois de muita discussão chegou-se a uma decisão. Os poços petrolíferos cujas coordenadas geográficas estiverem à direita da linha demarcatória definida pela ONU pertencerão ao Uzbequistão e, em caso contrário, serão explorados pelo Turcomenistão.

O seu objetivo nessa questão é escrever um programa C que solicite ao usuário a digitação na entrada padrão das coordenadas X e Y da localização de um poço petrolífero e que exiba na saída padrão o nome do país ao qual pertence. Lembre-se que só pode haver poços na região demarcada em cinza. Portanto, se as coordenadas digitadas corresponderem a um ponto fora dessa área, uma mensagem de erro POCO INEXISTENTE deverá ser exibida. Além disso, se o poço estiver exatamente na linha litigiosa, o poço será explorado pela ONU.

A entrada e a saída de dados devem seguir **rigorosamente** o exemplo de execução apresentado. Considere as coordenadas X e Y como números reais. Chame o programa de `questao1.c`

#### Exemplo de execução:

```
Poco (X,Y): 1.5 -2.5
Turcomenistao
```

(continua no verso)

### Questão 2 (\*\*\*): Fuso Horário

[30 pontos]

Você foi escolhido para participar de um grande projeto colaborativo mundial onde deve trocar informações com outros programadores residentes em diversos locais do planeta. Porém, sempre que é necessário ligar para um colega distante, você precisa verificar se o fuso horário é adequado para manter contato.

Como se sabe, existem 12 fusos atrasados e outros 12 fusos adiantados em relação ao horário de Brasília. Assim, para um fuso de -4h, o horário local de 16h45min corresponde a 12h45min. Por outro lado, um fuso de +10h corresponde a 2h45min, já na madrugada do dia seguinte. Para facilitar a sua vida, você resolveu escrever um programa C para ajudá-lo nessa tarefa.

O programa deve solicitar ao usuário o horário local e também a diferença de fuso horário em relação à região do planeta com a qual precisa se comunicar. A saída deve ser o horário correspondente no fuso horário fornecido como dado de entrada. A entrada e a saída devem seguir **rigorosamente** o exemplo de execução apresentado abaixo. Chame o programa de `questao2.c`

#### Exemplo de execução:

```
Hora atual (HH:MM): 16:45
Fuso horario.....: +10
Hora no fuso.....: 2h45min
```

---

### Questão 3 (\*): Teorema dos Cosenos

[40 pontos]

Em Geometria Plana, o teorema dos cosenos declara que se os comprimentos de dois dos lados de um triângulo e o ângulo entre eles são conhecidos, então o comprimento do terceiro lado pode ser calculado. Assim, dado o comprimento dos lados  $\overline{A}$  e  $\overline{B}$  de um triângulo e o ângulo  $\widehat{AB}$  entre eles, em graus, o terceiro lado, identificado por  $\overline{C}$ , é calculado segundo a expressão:

$$\overline{C}^2 = \overline{A}^2 + \overline{B}^2 - 2 \cdot \overline{B} \cdot \overline{C} \cdot \cos(\widehat{AB})$$

Seu objetivo nessa questão consiste em escrever um programa C que solicite ao usuário a digitação dos valores de  $\overline{A}$ ,  $\overline{B}$  e  $\widehat{AB}$  e que, em seguida, calcule e exiba na saída padrão o valor do lado  $\overline{C}$  com precisão de três dígitos decimais. Use números reais para todos os dados de entrada. Chame o programa de `questao3.c`

Lembre-se que para converter um ângulo em graus para radianos deve-se multiplicar o ângulo por  $\pi/180$ . O formato de saída deve ser **exatamente** conforme o exemplo de execução que se segue.

#### Exemplo de execução:

```
Lado A...: 10
Lado B...: 8
Angulo AB: 30
Lado C...: 5.043
```

---

### Questão 1: Mapa do Tesouro

[35 pontos]

As crianças acharam um *mapa do tesouro* contendo instruções do tipo “ande cinco passos para o norte” e “dê 2 passos para o oeste”. O tesouro será encontrado se as crianças conseguirem seguir corretamente as instruções contidas no mapa.

Seu objetivo nessa questão é escrever um programa que simule a brincadeira do mapa do tesouro, informando a direção e o número de passos a serem dados. Não é necessário fazer a consistência dos valores digitados pelo usuário. Chame o programa de `tesouro.c`

A entrada consiste em quatro linhas, cada uma contendo um caracter que identifique a direção (N para Norte, S para Sul, L para Leste ou O para Oeste) e um inteiro que represente o número de passos a serem dados. A saída corresponde às coordenadas no plano cartesiano referentes à posição onde se encontra o tesouro escondido. Considere a saída da brincadeira a partir das coordenadas (0,0) do plano cartesiano.

#### Exemplo de entrada e de saída:

```
Instrucao 1: N 3
Instrucao 2: L 2
Instrucao 3: O 5
Instrucao 4: S 1
Tesouro: (-3,2)
```

---

### Questão 2: Vaquinha

[35 pontos]

João, Pedro e Carlos fizeram uma vaquinha para irem ao cinema e lancharem no *McDonalds*. Cada um contribuiu com o dinheiro que tinha no bolso. No final, João ficou encarregado de somar todas as despesas e mostrar aos colegas quanto cada um deveria receber de volta ou pagar a mais de forma que todos contribuíssem exatamente com o mesmo valor.

Seu objetivo nesta questão é escrever um programa C que facilite o trabalho do João. Não é necessário fazer a consistência dos valores digitados pelo usuário. Chame o programa de `vaquinha.c`

A entrada consiste em duas linhas. A primeira linha deve conter um número real e positivo correspondente ao total das despesas. A segunda linha deve conter três números reais e positivos correspondentes, respectivamente, à quantidade de dinheiro fornecida por João, Pedro e Carlos.

A saída consiste em três linhas contendo cada uma um número real com precisão de duas casas decimais correspondente ao valor a ser pago ou a receber por cada um dos amigos.

**Exemplo de entrada e de saída:**

Total da despesa: 58.34

Vaquinha: 18 15 21

Joao deve receber 4.55

Pedro deve pagar 3.55

Carlos pagou a quantia correta

---

**Questão 3: Notas e Conceitos**

[30 pontos]

Os cursos de pós-graduação *stricto sensu* (mestrado e doutorado) no Brasil adotam um critério de pontuação através de conceitos. Há uma correlação entre as notas finais obtidas pelo aluno e o conceito a ser lançado no histórico escolar, conforme apresentado na tabela que se segue:

Nota Final	Conceito
De 100 a 85	A
De 84 a 70	B
De 69 a 50	C
Inferior a 50	D

Seu objetivo nesta questão é escrever um programa C que solicite ao usuário a nota final de um aluno e que exiba na tela o conceito correspondente. Se o usuário digitar uma nota fora do intervalo  $[0, 100]$  uma mensagem de erro deve ser apresentada na tela. Chame o programa de `conceito.c`

A entrada consiste em uma única linha contendo um número inteiro correspondente à nota final do aluno. A saída consiste em uma única linha contendo o conceito obtido pelo aluno.

**Exemplo de entrada e de saída:**

Nota final: 72

Conceito: B

---

	Qt.1	Qt.2	Qt.3	Total
Valor	35	35	30	100
Pontos obtidos				