Lista de Exercícios - Condicionais Parte I

1. OBI - Exercício do Bondinho. A turma do colégio vai fazer uma excursão na serra e todos os alunos e monitores vão tomar um bondinho para subir até o pico de uma montanha. A cabine do bondinho pode levar 50 pessoas no máximo, contando alunos e monitores, durante uma viagem até o pico. Neste problema, dado como entrada o número de alunos e o núnero de monitores, você deve implementar um programa que diga se é possível ou não levar todos os alunos e monitores em apenas uma viagem

2. Faça um programa que tendo como entradas (via teclado) a base e altura de um retângulo, calcule o perímetro (2*base + 2*altura) e a área (base*altura) e imprima se o perímetro é maior que a área.

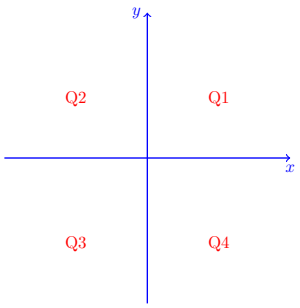
3. Modifique o programa anterior para que imprima qual dos dois é maior (perímetro ou área). Assuma que nunca podem ser iguais.

4. Modifique o programa anterior para que imprima qual dos dois é maior (perímetro ou área) ou se são iguais.

5. Faça um programa que dados os coeficientes (a,b e c) de uma equação do 2o grau, calcule e imprima suas raízes (caso a equação possua **raízes imaginárias**, o programa **não deve imprimir nada**).

6. Refaça o exercício anterior para imprimir mensagem `raízes imaginárias` caso as raízes sejam imaginárias. Caso a equação possua uma única raíz o programa também deve indicar isso (ou seja que a raiz é única).

7. Leia 2 valores reais (x e y), os quais representam as coordenadas de um ponto em um plano. A seguir, determine a qual quadrante pertence o ponto. Analise também se está sobre um dos eixos cartesianos ou na origem ($x = y = 0$). Para auaxiliar a resoluação do exercícico, a figura abaixo ilustra os quatro quadrantes no plano cartesiano $X \times Y$, como primeiro quadrante iniciando no canto superior direito, o segundo no canto superior esquerdo e assim por diante.



Exemplo de Entrada	Exemplo de Saída
4.5 -2.2	Q4
0.1 0.1	Q1
0.0 0.0	Origem
0.1 0.0	Eixo x

.

8. Dadas a data atual e a data de nascimento de uma pessoa,

   - (a) calcular a sua idade.

     Complete o programa do item anterior, informando também

   - (b) o dia da semana em que a pessoa nasceu
     Dica: uma possibilidade é `https://pt.wikipedia.org/wiki/Congruência_de_Zeller`

Da wiki temos:

Para um calendário gregoriano, a congruência de Zeller ajustada para $\mod$ como resto da divisão ($\mod \leftrightarrow \%$) é

$$h = \left(q + \left\lfloor \frac{(m+1)26}{10} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J\right) \mod 7$$

em que

- h é o dia da semana (0 = sábado, 1 = domingo, 2 = segunda, ...)
- q é o dia do mês
- m é o mês (3 = março, 4 = abril, 5 = maio,...)
- K é o ano do século ($ano \mod 100$)
- J é o século ($\lfloor ano/100 \rfloor$ (por exemplo, para 1995 o século seria 19, ainda que na realidade o século seria XX)

Nota 1: neste algoritmo, janeiro e fevereiro são contados como os meses 13 e 14 do ano anterior.

Portanto, na equação anterior, mod $\leftrightarrow$ % e $\lfloor\ \rfloor$ representa o truncamento da divisão (divisão inteira).

9. Desafio: Faça um programa que:

   - Gere dois números aleatórios, `n1` e `n2`, inteiros positivos e no intervalo $[1, 100]$
   - Escolha aleatoriamente uma das operações aritméticas a seguir:
     soma, subtração, multiplicação, quociente da divisão (parte inteira da divisão), resto da divisão.
   - Peça ao usuário que forneça o resultado da operação aritmética escolhida considerando os valores de `n1` e `n2`.
   - Monitore o tempo de resposta do usuário (dica: use uma função da biblioteca `time.h`).
   - Exiba uma mensagem dizendo se o usuário acertou ou errou a resposta, acompanhada do tempo calculado.

   Dica: use a função `rand()`. Não se esqueça da inicialização da semente: `srand(time(NULL))`.

10. Considere o código abaixo:

```c
#include <stdio.h>

int main ()
{
    int n1, n2, n3, aux;

    if (n2 >= n3)
    {
        aux = n2;
        n2 = n3;
        n3 = aux;
    }
    if (n1 >= n2)
    {
        aux = n1;
        n1 = n2;
        n2 = aux;

        if (n2 >= n3)
        {
            aux = n2;
            n2 = n3;
            n3 = aux;
        }
    }
    printf ("%d %d %d\n", n1, n2, n3);
    return (0);
}
```

Sem executar o programa, responda:

- Qual será a saída se atribuirmos no início os valores `1`, `2` e `3` a `n1`, `n2` e `n3`, respectivamente?
- Qual será a saída se atribuirmos no início os valores `20`, `10` e `30` a `n1`, `n2` e `n3`, respectivamente?
- Qual será a saída se atribuirmos no início os valores `5*5`, `n1/2` e `n2+1` a `n1`, `n2` e `n3`, respectivamente?
- O que o programa faz, exatamente? Explique o funcionamento do algoritmo.
- Para carregar a resposta a este exercício, transcreva o código fonte acima e indique as respostas como comentarios // ou /*...*/ num arquivo do codeblocks ou um editor de texto (txt) qualquer.