



## Lista 3 - Estrutura de Decisão

### Exercícios para sala de aula

1) O código a seguir tem o objetivo de obter o resto da divisão do número maior pelo menor informados (independentemente da ordem informada) e informar que não é possível realizar a divisão caso o divisor seja zero. Faça os ajustes necessários no código para que esses objetivos sejam alcançados.

```
#include <stdio>

int main(void)
{
    float num1;
    int num2, resto;

    printf("Informe o primeiro valor: ");
    scanf("%d", num1);
    printf("Informe o segundo valor: ")
    scanf("%d", num2);

    if(num1 > num2)
    {
        resto = num1 % num2;
    }
    else
    {
        resto = num2 % num1;
    }
    else if(num1 || num2 == 0)
    {
        printf("Nao eh possivel realizar divisao por zero\n");
    }

    printf("O resto da divisao eh %f\n", resto);

    return 0;
}
```

2) Ler dois números inteiros e informar:

a) Se ambos são divisíveis por 5.

- b) Se pelo menos um deles é divisível por 5.
- c) Se ambos são pares.
- d) Se pelo menos um deles é ímpar.

3) Elabore um programa que leia o dia e o mês de nascimento de uma pessoa e determine o seu signo conforme a tabela a seguir:

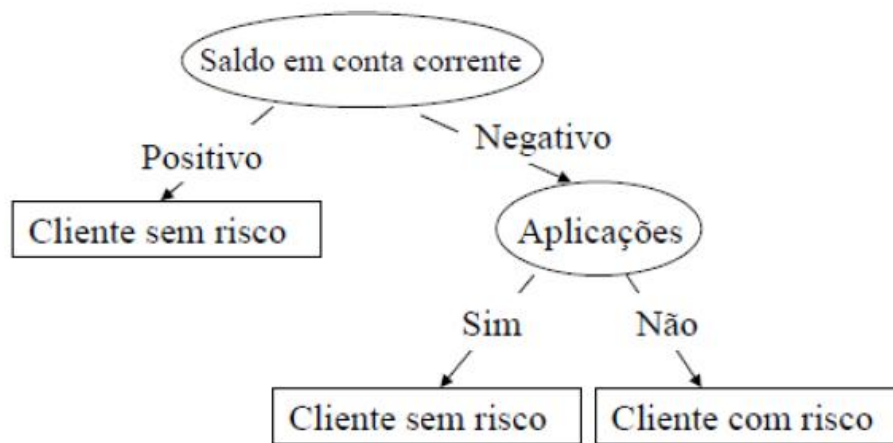
Intervalo	Signo
de 22/12 até 20/1	Capricórnio
de 21/1 até 19/2	Aquário
de 20/2 até 20/3	Peixes
de 21/3 até 20/4	Áries
de 21/4 até 20/5	Touro
de 21/5 até 20/6	Gêmeos
de 21/6 até 21/7	Câncer
de 22/7 até 22/8	Leão
de 23/8 até 22/9	Virgem
de 23/9 até 22/10	Libra
de 23/10 até 21/11	Escorpião
de 22/11 até 21/12	Sagitário

Se informada uma data que não corresponde aos intervalos indicados, informar que a data é inválida.

4) Um ano é bissexto se for divisível por 4 e não for divisível por 100. Também são bissextos os divisíveis por 400. Escreva um programa que determina se um ano informado pelo usuário é bissexto.

5) Uma árvore de decisão obtém a decisão pela execução de uma sequência de testes. Cada nó interno da árvore corresponde a um teste do valor de uma das propriedades e os ramos deste nó são identificados com os possíveis valores do teste. Cada nó folha da árvore especifica o valor de retorno se a folha for atingida.

5.1) A figura a seguir apresenta um exemplo fictício de árvore de decisão, tomando atributos de clientes de uma instituição financeira. Elabore um programa que implemente essa árvore de decisão. As entradas podem ser do tipo *char*, portanto, considere a primeira letra de cada palavra como entrada. Caso o usuário informe um caractere diferente dos aceitáveis, apresente a mensagem "*Caractere inválido*".



5.2) Considera-se o problema de esperar para jantar em um restaurante. O objetivo é o programa aprender a definição para *DeveEsperar?* Existem os seguintes atributos para descrever as situações:

**Fregueses:** Quantas pessoas estão no restaurante (nenhuma, algumas, cheio)?

**Tempo de espera (em minutos):** 0-10, 10-30, 30-60, >60.

**Alternativa:** Há um restaurante alternativo na redondeza?

**Reserva:** Foi feita uma reserva?

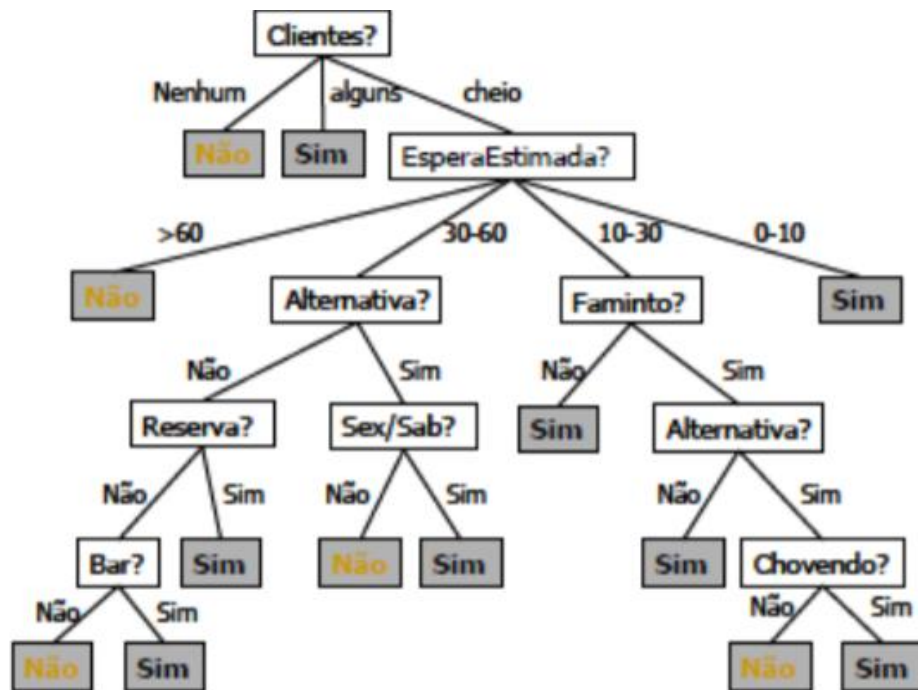
**Bar:** Existe um bar confortável onde possa se esperar?

**Sex/Sab:** É sexta ou sábado?

**Faminto:** Está com fome?

**Chovendo:** Está chovendo lá fora?

As entradas são do tipo *char*, exceto o tempo de espera que pode ser *int*. Portanto, estabeleça uma letra para cada entrada esperada, por exemplo 'S' para sim e 'N' para não.



6) Crie um programa que leia dia, mês e ano separadamente e imprima se a data é válida ou não.

7) Crie um programa que leia uma data no formato **ddmmaaaa** e imprima se a data é válida ou não.

8) Implemente um programa que adivinhe o "número mágico" entre 0 e 10. O programa deverá imprimir a mensagem *"Certo! %d é o número mágico"* quando o jogador acerta o número mágico, a mensagem *"Errado, muito alto"*, caso o jogador tenha digitado um número maior que o número mágico e a mensagem *"Errado, muito baixo"*, caso o jogador tenha digitado um número menor que o número mágico. O número mágico é produzido usando o gerador de números randômicos de C (função *rand()*, que exige o uso da biblioteca *stdlib.h*).