ATIVIDADE 2: STRUCTS/REGISTROS/RECORDS

Caríssimos(as) alunos(as), *esta atividade tem como objetivo reforçar o aprendizado sobre a definição, criação e utilização de STRUCT em C++. Resolva com atenção e dedicação, a fim de que seja mais proveitosa a nova disciplina. Utilize o Code Blocks para realizar a atividade. Crie uma pasta chamada: Ativ2-SEUNOME e salve dentro dela os arquivos criados em cada questão (com o nome qN.cpp, onde N é o número da questão). Ao final, compacte a pasta e envie-a na tarefa criada na sala virtual da disciplina. Bons estudos!*

---

1. Você participará de uma equipe que desenvolverá um programa para informatizar um banco, fazendo o controle dos clientes e de suas contas. Neste primeiro momento, serão definidas apenas as estruturas básicas para o cadastro de um cliente e sua respectiva conta. Faça um programa que:
   a) declare os registros **TCliente** e **TConta**, conforme descrição a seguir:
      i. os dados de Cliente são: **matrícula**, **nome**, **idade** e **conta**;
      ii. os dados da Conta são: **número**, **tipo** (Poupança ou Conta Corrente) e **saldo**;
   b) crie todas as variáveis necessárias para realizar o cadastro de **01 (um) cliente** e sua conta;
   c) crie um procedimento para realizar o cadastro;
   d) crie um procedimento para mostrar os dados cadastrados.

---

2. Salve o exercício anterior com outro nome e altere-o para que permita o cadastro de até 10 clientes.

---

3. Construa um programa que possua uma estrutura para representar os alunos de um determinado curso. A estrutura deve conter: matrícula do aluno (**inteira e autoincrementada**), **nome**, e as **notas** de **três avaliações**. O programa deve ser capaz de:
   a) permitir a leitura e armazenamento em memória dos dados de até 5 alunos;
   b) retornar a matrícula e o nome do aluno com maior nota da primeira avaliação;
   c) retornar todos os dados do aluno com maior média das notas;
   d) retornar o resultado final da disciplina, contendo a matrícula, o nome e o status do aluno (aprovado: média >= 6 pts; reprovado, caso contrário)

---

4. Faça um programa que leia duas datas e armazene-as em estruturas definidas por:

```
struct minhaData{
        int dia;
        int mes;
        int ano;
};
```

   Em seguida, calcule o número de dias entre as duas datas. OBS: a primeira data deve ser anterior que a segunda.

---

5. Implemente uma Agenda que permita o armazenamento de compromissos conforme especificação a seguir:
   a) a principal entidade do programa é o **Evento**, o qual é composto por um **nome**, uma **data**, um **horário de início**; um **horário de término** e a **descrição** do evento;
   b) o **horário** deve armazenar a hora e os minutos;
   c) a **data** deve armazenar o dia, o mês e o ano;

   Demonstre o funcionamento do programa permitindo o armazenamento de até 10 compromissos.

---

6. Construa um programa que gerencie o consumo de energia dos eletrodomésticos de uma residência. Para isso, deverão ser cadastrados **até 10** eletrodomésticos, coletando os seguintes dados para cada um: **nome, potência (em Watts) e tempo de uso por dia (em horas)**.
   Em seguida, permita ao usuário informar a quantidade de dias que deseja verificar o consumo. Mostre a seguinte saída:
   a) Consumo total em **kWatts** (que é a medida usada na conta de energia);
   b) Consumo individual de cada eletrodoméstico, também em kWatts;
   c) Consumo relativo (percentual) de cada eletrodoméstico em relação ao consumo total da residência.