Leia com atenção o enunciado apresentado por cada exercício, e desenvolva o algoritmo solicitado.

1) Implemente uma função para somar variáveis do tipo horário, conforme a struct abaixo. Você deve implementar ainda um main() para entrar com valores de dois horários e usar a função de soma que você implementou.

Ex: 10:45:35 + 5:20:40 = 16:06:15

```
struct horario {
  int hora, min, seg;
};
```

2) Elabore um programa em C que leia do usuário uma data (criar o tipo data, com dia, mês e ano). Para a verificação da data deve ser feito uma função que receba a data e retorne verdade se ela é válida e falso caso contrário. Anos bissextos são dados pelas regras (segundo o calendário Gregoriano):

i) De 4 em 4 anos é ano bissexto.

ii) De 100 em 100 anos não é ano bissexto.

iii) De 400 em 400 anos é ano bissexto.

iv) Prevalecem as últimas regras sobre as primeiras.

A título de curiosidade, o ano de 1900 foi o último ano a ser aplicada a regra ii (não é bissexto). A próxima vez será em 2100.

3) Escreva uma função que receba dois structs do tipo dia, mês e ano, cada um representando uma data válida (utilize o algoritmo feito no exercício 2), e devolva o número de dias que decorreram entre as duas datas.

4) Faça um programa que, usando struct, armazene o nome e a data de nascimento de até 10 pessoas (o usuário entrará com estas informações). A geração da data de nascimento deve ser feita aleatoriamente através da função abaixo:

```
void CriaData (TData &D) {
  D.Mes = 1 + (rand() % 12);
  D.Ano = 1940 + (rand() % 74);
  D.Dia = 1 + (rand() % 30);
}
```

O programa deve:

- Verificar se a data de nascimento gerada é válida (utilize o algoritmo desenvolvido na exercício número 2);
- listar todos os nomes e respectivas idades; e
- listar os nomes das pessoas mais velhas do que uma certa idade (deve ser validada também esta data) fornecida pelo usuário.

Devem ser criadas funções diferentes para cada uma dessas atividades.

5) Faça um programa de controle de despesas e dados de um condomínio de apartamentos. Os dados de cada apartamento são armazenados em um vetor de estruturas. Para cada apartamento tem-se os seguintes dados:

- Nome do responsável.
- Número do apartamento.
- Área em m^2.
- Número de moradores.
- Valor a ser pago no mês.

Crie a estrutura descrita acima e declare um vetor de estruturas, lembrando que o condomínio possui 40 apartamento. A seguir, crie um menu no main e:

- Crie uma função que preencha o vetor acima, exceto o campo valor.
- Informe a área total do condomínio.
- Informe qual apartamento possui o maior número de moradores, e imprima seus dados. Caso haja mais que um, imprima as informações de todos.
- Tenha uma opção para sair do programa.