1) Defina o tipo horario, com os campos hora, minuto e segundo. Faça uma função que recebe por parâmetro o tempo de duração de uma prova e retorna o mesmo tempo em segundos.

   int calculaTempo(horario t);

2) Defina o tipo horario, com os campos hora e minuto. Faça uma função que recebe, por parâmetro, o horário de início e o horário de término de um jogo, ambos subdivididos em 2 valores distintos: horas e minutos. A função deve retornar a duração do jogo em minutos, considerando que o tempo máximo de duração de um jogo é de 24 horas e que o jogo pode começar em um dia e terminar no outro.

   int duracao(horario ini, horario fin)

3) Faça uma função que recebe como parâmetro um valor n, inteiro e positivo, calcule e retorne o resultado da seguinte soma:

   $S = 1 - 1/2 + 1/3 - 1/4 + ... + 1/n$

4) Escreva uma função que imprima todos os números perfeitos existentes entre dois números, chamada imprimePerfeitos. Observe que essa função deve imprimir um número apenas se ele for perfeito. Para descobrir se um número é perfeito você pode chamar a função ehPerfeito implementada por você. A função ehPerfeito retorna 1 se o número é perfeito e 0 caso contrário. Lembre-se, um número é dito perfeito quando ele é igual a soma dos seus divisores excetuando ele próprio. (Ex: 6 é perfeito, 6 = 1 + 2 + 3, que são seus divisores).

5) Dado um conjunto com n elementos, chama-se arranjo simples de taxa k, a todo agrupamento de k elementos distintos dispostos numa certa ordem. Faça uma função para calcular o número total de arranjos de n elementos tomados k a k através da fórmula:

   $$A_{n,k} = n! / (n-k)!$$

   Esta função deve obrigatoriamente chamar uma função fatorial implementada por você com o seguinte protótipo: int fatotial (int n)

   Escreva também o main para testar a função arranjo.

6) Escreva um programa que leia pares de valores positivos (use a função leNaoNegativo). Imprima se os elementos de cada par são números amigos (ou não). A leitura dos pares deve terminar quando o usuário digitar o par 0 e 0. Dois números A e B são amigos se a soma dos divisores de A excluindo A é igual a B e a soma dos divisores de B excluindo B é igual a A. Para a verificar se dois números são amigos utilize a função saoAmigos. A função saoAmigos deve usar a função somaDivisores.

   Nome: leNaoNegativo
   Objetivo: Ler um valor não negativo. Se o número for negativo, a leitura deve ser repetida até que o valor lido seja zero ou positivo.
   Parâmetros: Nenhum.
   Retorno: (int) o valor não negativo.

   Nome: saoAmigos
   Objetivo: verifica se dois números n1 e n2 são amigos.
   Parâmetros: (int,int) n1 e n2 (inteiros positivos).
   Retorno: (int) 1 se os dois números são amigos, 0 caso contrário.
   Observação: Utilize a função somaDividores
   Exemplo:
   220 e 284 são amigos, pois
   220: 1+2+4+5+10+11+20+22+44+55+110=284
   284: 1+2+4+71+142=220
   1184 e 1210 também são amigos.

   Nome: somaDivisores
   Objetivo: Calcula a soma dos divisores de um número(exceto ele mesmo) n.
   Parâmetros: O número n (inteiro e positivo).
   Retorno: A soma dos divisores de n.
   Exemplo: Para o valor 8: 1+2+4 = 7