

1.  
/\* O algoritmo para este exercício é MUITO mais simples do que o enunciado faz parecer. Apesar de isso parecer um "golpe baixo", este tipo de situação é extremamente comum – os problemas reais sempre envolvem reflexão sobre o que é relevante e o que faz parte da solução do problema. Acostume-se com enunciados, especificações e descrições com bastante texto: eles acompanharão você para sempre. \*/

```
#include <stdio.h>
```

```
int main ()
{
    double preco_a, preco_g; // Preço, em reais/litro
    double rendimento_a, rendimento_g; // Rendimento, em km/litro

    printf ("Preço do alcool (por litro): ");
    scanf ("%lf", &preco_a);
    printf ("Preço da gasolina (por litro): ");
    scanf ("%lf", &preco_g);
    printf ("Rendimento com alcool (km/litro): ");
    scanf ("%lf", &rendimento_a);
    printf ("Rendimento com gasolina (km/litro): ");
    scanf ("%lf", &rendimento_g);

    double reais_por_km_alcool = preco_a / rendimento_a; // reais/km
    double reais_por_km_gasolina = preco_g / rendimento_g; // reais/km

    if (reais_por_km_alcool < reais_por_km_gasolina)
        printf ("Melhor usar alcool.\n");
    else
        printf ("Melhor usar gasolina.\n");

    return (0);
}
```

---

2.  
/\* Como sabemos que são 3 dígitos, podemos usar a mesma estratégia que já tínhamos usado para separar um número de segundos em horas, minutos e segundos – aqui, separamos as unidades em centenas, dezenas e unidades. Repare também que o problema não exige o uso da estrutura condicional! \*/

```
#include <stdio.h>
```

```
int main ()
{
    int n, c, d, u;
    scanf ("%d", &n);

    c = n/100; // Centenas = quociente da divisão por 100.
    d = (n%100)/10; // Dezenas = o resto da divisão por 100 dividido por 10.
    u = (n%10); // Unidades = o que não for divisível por 10.

    printf ("%d\n", c + d*10 + u*100); /* Imprimimos na ordem contrária. */
    return (0);
}
```

3. Vou dar aqui 2 soluções. Note como elas são até certo ponto análogas às soluções dadas para o problema das pepitas de ouro na 1a lista.

/\* Solução 1: compara 2 valores, faz uma segunda comparação dependendo do resultado da primeira comparação. \*/

```
#include <stdio.h>
```

```
int main ()
{
    int n1, n2, n3;
    scanf ("%d %d %d", &n1, &n2, &n3);

    if (n1 > n2)
    {
        if (n1 > n3)
            printf ("%d\n", n1);
        else
            printf ("%d\n", n3);
    }
    else
    {
        if (n2 > n3)
            printf ("%d\n", n2);
        else
            printf ("%d\n", n3);
    }

    return (0);
}
```

/\* Solução 2: guarda em uma variável n1, e compara esta variável com n2, e depois com n3. Esta solução é estruturalmente mais simples, e é mais parecida com a solução para o problema mais geral, que teria um número arbitrário de valores. \*/

```
#include <stdio.h>
```

```
int main ()
{
    int n1, n2, n3, maior;

    scanf ("%d %d %d", &n1, &n2, &n3);
    maior = n1;

    if (n2 > maior)
        maior = n2;

    if (n3 > maior)
        maior = n3;

    printf ("%d\n", maior);
    return (0);
}
```

4.

```
/* Este problema parece mais complexo, mas na verdade a única dificuldade é
   como lidar com a "virada" do dia. */
```

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int inicio, fim;
```

```
    printf ("Hora inicial e final: ");
```

```
    scanf ("%d %d", &inicio, &fim);
```

```
    /* Se a hora de fim for menor que a hora de início, o jogo terminou no
       dia seguinte. */
```

```
    if (fim <= inicio)
```

```
        fim = fim + 24;
```

```
    printf ("O jogo durou %d hora(s).\n", fim - inicio);
```

```
}
```

```
/* Outra possível solução. */
```

```
int main ()
```

```
{
```

```
    int inicio, fim, duracao;
```

```
    printf ("Hora inicial e final: ");
```

```
    scanf ("%d %d", &inicio, &fim);
```

```
    /* Se a hora de fim for menor que a hora de início, o jogo terminou no
       dia seguinte. */
```

```
    if (inicio < fim)
```

```
        duracao = fim - inicio;
```

```
    else
```

```
        duracao = 24 - inicio + fim ;
```

```
    printf ("O jogo durou %d hora(s).\n", duracao);
```

```
}
```

```

/* E para o desafio! Aqui fica mais complicado... */

# include <stdio.h>

int main ()
{
    int horaInicial, minutoInicial,
        horaFinal, minutoFinal ,
        duracaoHoras, duracaoMinutos;

    printf ("Digite horaInicial minutoInicial horaFinal minutoFinal: ");
    scanf ("%d %d %d %d", &horaInicial, &minutoInicial, &horaFinal, &minutoFinal);

    // Duracao em horas (lógica do exercício anterior).
    if (horaInicial < horaFinal)
        duracaoHoras = horaFinal - horaInicial;
    else
        duracaoHoras = 24 - horaInicial + horaFinal;

    /* ATENCAO : note que aqui temos um bloco if/else separado do anterior! São
    duas estruturas condicionais INDEPENDENTES! */
    if (minutoInicial <= minutoFinal)
        duracaoMinutos = minutoFinal - minutoInicial;
    else
    {
        // Subtrair de 60 segue a mesma lógica das horas.
        duracaoMinutos = 60 - minutoInicial + minutoFinal ;

        // Você consegue justificar por que este ajuste é necessario?
        duracaoHoras = duracaoHoras - 1;
    }

    printf ("O jogo teve duracao de %d hora(s) e %d minuto(s)\n", duracaoHoras,
        duracaoMinutos);

    return 0;
}

```

---

5.

a) 1 2 3

b) 10 20 30

c) O programa reordena  $n_1$ ,  $n_2$  e  $n_3$  de forma que eles estejam sempre em ordem crescente. Este algoritmo funciona com trocas. O primeiro if garante que  $n_2$  terá valor menor ou igual a  $n_3$ . O segundo if compara  $n_1$  com este valor. Se  $n_1 \leq n_2$ , sabemos que  $n_2 \leq n_3$ , portanto  $n_1 \leq n_3$  e os valores já estão ordenados. Do contrário, trocamos os valores de  $n_1$  e  $n_2$ , garantindo que o primeiro valor é o menor de todos. Precisamos então comparar novamente  $n_2$  com  $n_3$ , para verificar se o valor que originalmente estava em  $n_1$  e agora está em  $n_2$  é maior que o valor em  $n_3$ . Para enxergar o funcionamento do algoritmo, use os valores 1, 2 e 3, em todas as ordens possíveis (são 6 casos).

6.

```
#include <stdio.h>

int main ()
{
    float x, y;

    printf ("Digite as coordenadas: ");
    scanf ("%f %f", &x, &y);

    if (x < 0) // Quando x é positivo.
    {
        if (y < 0)
            printf ("Q3\n");
        else if (y == 0)
            printf ("Eixo X\n");
        else
            printf ("Q2\n");
    }
    else if (x == 0) // Quando x é zero!
    {
        if (y == 0)
            printf ("Origem \n");
        else
            printf ("Eixo Y\n");
    }
    else // Quando x é negativo.
    {
        if (y < 0)
            printf ("Q4\n");
        else if (y == 0)
            printf ("Eixo X\n");
        else
            printf ("Q1\n");
    }

    return 0;
}
```