

1.

Alterar a linha 5 para: `unsigned long long total = 1;`
Alterar a linha 12 para: `total *= base;`

2.

```
#include <stdio.h>

#define PONTOS_VENCEDOR 10 // Pontos ganhos por adivinhar o vencedor.
#define PONTOS_PLACAR 5 // Pontos ganhos por adivinhar o placar de um time.

int main ()
{
    int aposta_t1, aposta_t2; // Número de gols dos times na aposta.
    int placar_t1, placar_t2; // Número de gols dos times no placar real.

    int pontos = 0;

    printf ("Informe o placar da aposta (ex: 2x1): ");
    scanf ("%dx%d", &aposta_t1, &aposta_t2);
    printf ("Informe o placar real (ex: 2x1): ");
    scanf ("%dx%d", &placar_t1, &placar_t2);

    // Verifica se a aposta identificou o vencedor.
    if (((aposta_t1 > aposta_t2) && (placar_t1 > placar_t2)) ||
        ((aposta_t1 < aposta_t2) && (placar_t1 < placar_t2)) ||
        ((aposta_t1 == aposta_t2) && (placar_t1 == placar_t2)))
        pontos += PONTOS_VENCEDOR;

    // Verifica se a aposta acertou o número de gols dos times.
    if (aposta_t1 == placar_t1)
        pontos += PONTOS_PLACAR;

    if (aposta_t2 == placar_t2)
        pontos += PONTOS_PLACAR;

    printf ("%d pontos\n", pontos);

    return (0);
}
```

3.

/* O problema aqui é descobrir na verdade um ingresso cujo número n é igual à sua posição na sequência i. Uma variante da solução abaixo (que não bate totalmente com a especificação) seria não ler todos os valores, interrompendo a execução imediatamente ao encontrar o vencedor. Isso poderia ser feito com uma flag, por exemplo. */

```
int main ()
{
    int n_ingressos, i, n;
    int vencedor = 0;

    // Inicializei n_vencedor com 0 para saber se, por erro, não teve vencedor!

    // Lê o número de ingressos.
    scanf ("%d", &n_ingressos);

    // Note que aqui começamos a contagem em 1 e vamos até n_ingressos.
    for (i = 1; i <= n_ingressos; i++)
    {
        // Lê o número do i-ésimo ingresso.
        scanf ("%d", &n);

        // Achou o premiado!
        if (n == i)
            vencedor = i;
    }

    // Mostra o vencedor.
    printf ("Vencedor: %d\n", vencedor);

    return (0);
}
```

4.

/* Este problema é uma "pegadinha". Em um primeiro momento, parece que precisa contar volta a volta até ver o resto. Mas não, você só precisa colocar o resto da divisão!!! Vou colocar ambas as versões aqui, mas a resposta "certa" é a mais eficiente. Se não conseguiu entendê-la, procure ajuda! */

```
#include <stdio.h>

int main ()
{
    int meta, tam_pista;

    scanf ("%d %d", &meta, &tam_pista);

    // Solução eficiente: é só um resto da divisão!!!
    printf ("%d\n", meta % tam_pista);

    // Solução ineficiente: dá voltas!
    // Para isso, vai descontando da meta o tamanho da pista.
    // Note que não preciso saber quantas voltas foram dadas, só o tamanho da
    // volta incompleta.
    while (meta >= tam_pista)
        meta -= tam_pista;
    printf ("%d\n", meta);

    return (0);
}
```