

1.

```
1xxxxxxxxxxxxx0
11xxxxxxxxxxxx00
111xxxxxxxxx000
1111xxxxxxx0000
11111xxxx00000
111111xx000000
1111111x000000
111111110000000
```

2.

/* A ideia é bem simples: basta percorrer a string até a metade, trocando o valor na posição i com aquele na posição tamanho-1-i (= de trás para a frente). Só pode ir até a metade, senão vai inverter e DESINVERTER! */

```
void inverteString (char* string)
{
    int tam = 0;
    int i;
    char aux;

    // Acha o tamanho.
    tam = 0;
    while (string [tam] != '\0')
        tam++;

    for (i = 0; i < tam/2; i++)
    {
        aux = string [i];
        string [i] = string [tam-i-1];
        string [tam-i-1] = aux;
    }
}
```

3.

```
/* Função auxiliar, diz se um caractere dado é uma letra minúscula. */
int ehLetraMinuscula (char c) {return ((c >= 'a') && (c <= 'z'));}

/* Função auxiliar, diz se um caractere dado é uma letra maiúscula. */
int ehLetraMaiuscula (char c) {return ((c >= 'A') && (c <= 'Z'));}

#define N_LETRAS 26 /* Existem 26 letras! */

/* c-offset dá a posição de um caractere c no vetor.
   Por exemplo, 'A'-'A' = 0, 'b'-'a' = 1, etc. */
#define OFFSET_MAIUSCULAS 'A'
#define OFFSET_MINUSCULAS 'a'

/* O truque é usar um vetor para contar cada uma das letras. */
void contaLetras (char* s)
{
    int i;
    int count [N_LETRAS];
    int size = strlen (s); /* Supondo uma string bem formada. */

    /* Começa zerando o vetor. */
    for (i = 0; i < N_LETRAS; i++)
        count [i] = 0;

    /* Outro jeito de fazer isso é usando a função memset da biblioteca-padrão.
       Note o cast para (void*). Funções que recebem este tipo costumam ser
       genéricas, funcionando para qualquer tipo de dado. */
    // memset ((void*) count, 0, (sizeof (int) * N_LETRAS));

    for (i = 0; i < size; i++)
    {
        if (ehLetraMinuscula (s[i]))
            count [s[i] - OFFSET_MINUSCULAS]++;
        else if (ehLetraMaiuscula (s[i]))
            count [s[i] - OFFSET_MAIUSCULAS]++;
    }

    /* Imprime a contagem para cada letra. */
    for (i = 0; i < N_LETRAS; i++)
        printf ("%c: %d\n", i + OFFSET_MAIUSCULAS, count [i]);
}
```