

Repetições - Operações Bit a Bit

Atividade 1 - Extrair bits de constantes hexadecimais

Objetivo

O objetivo dessa atividade é explorar as operações da linguagem C que podem ser utilizadas para "varrer" bits de uma variável.

Problema

Considere o seguinte programa:

```
1 #include <stdio.h>
2 int main() {
3     unsigned char i, mask, letter[7] = {0x10, 0x38, 0x44, 0x44, 0x7C, 0x44, 0x44};
4     for (i = 0; i < 7; i++) {
5         for (mask = 0x80; mask; mask >>= 1)
6             printf("%c", letter[i] & mask ? '*' : ' ');
7         printf("\n");
8     }
9 }
```

A cada passo das repetições, as seguintes operações são executadas:

i	letter[i]	mask	letter[i] & mask	printf
0	0x10 ₁₆	0x80 ₁₆	00010000 ₂ & 10000000 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x40 ₁₆	00010000 ₂ & 01000000 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x20 ₁₆	00010000 ₂ & 00100000 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x10 ₁₆	00010000 ₂ & 00010000 ₂ = 00010000 ₂	'*'
	0x10 ₁₆	0x08 ₁₆	00010000 ₂ & 00001000 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x04 ₁₆	00010000 ₂ & 00000100 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x02 ₁₆	00010000 ₂ & 00000010 ₂ = 00000000 ₂	' '
	0x10 ₁₆	0x01 ₁₆	00010000 ₂ & 00000001 ₂ = 00000000 ₂	' ' e '\n'
1	0x38 ₁₆	0x80 ₁₆	00111000 ₂ & 10000000 ₂ = 00000000 ₂	' '
	0x38 ₁₆	0x40 ₁₆	00111000 ₂ & 01000000 ₂ = 00000000 ₂	' '
	0x38 ₁₆	0x20 ₁₆	00111000 ₂ & 00100000 ₂ = 00100000 ₂	'*'
	0x38 ₁₆	0x10 ₁₆	00111000 ₂ & 00010000 ₂ = 00010000 ₂	'*'

...

Com a execução desse programa, aparecerá no terminal o seguinte resultado:

```
*
***
*  *
*  *
*****
*  *
*  *
```

Pode-se observar que o "desenho" da letra 'A' está nos bits dos elementos do vetor `letter` (os valores em `letter` estão em hexadecimal para facilitar a conversão para binário):

i	letter[i]	binário	binário-texto ('.' = 0 e '*' = 1)
0	0x10	00010000	...*....
1	0x38	00111000	..***...
2	0x44	01000100	.*...*..
3	0x44	01000100	.*...*..
4	0x7C	01111100	.*****..
5	0x44	01000100	.*...*..
6	0x44	01000100	.*...*..

O que este programa faz é varrer esses bits, usando as operações bit a bit da linguagem C `&` (e bit-a-bit) e `>>` (deslocamento à direita), para escrever na tela espaço ou `*` e saltar linhas para restituir a forma da letra que está armazenada em binário nesse vetor.

Modifique o programa para receber o número de letras n e em sequência uma matriz ($7 \times n$, ou seja sete linhas e n colunas) que contém uma mensagem codificada em binário. Cada coluna dessa matriz de entrada representa uma letra/símbolo da mensagem, por exemplo:

Para entrada:

```
5
112  0  16  0  0
 32  0   0  0 24
 32 68 48 124 60
 32 68 16  72 24
 34 68 16  16 24
 38 68 16  36  0
126 58 56 124 24
```

A saída será:

```
***                *
*
*                **
*      *  *  **  *****  ****
*      *  *  *   *   *  *   **
*  *  *  *  *   *       *   **
* ** *  *  *   *   *  *
*****  *** *  ***  *****  **
```

Descrição

1. Modificar o programa para realizar a leitura do número de colunas e dos dados da matriz e apresentar a mensagem conforme explicado. Considere que a matriz não terá mais do que 20 colunas.
2. NÃO acrescentar comandos de escrita precedendo as leituras como, por exemplo:

```
printf("\nDigite o número de colunas da matriz: ");
```

Esses comandos não são necessários porque a correção do programa será realizada automaticamente, com um script que fornecerá os valores do programa.

3. Abaixo algumas entradas para testar no programa: