



Lista de Exercícios de HW: operadores bit-a-bit e porta paralela

Obs: tente fazer no compilador esse exercício até conseguir. Caso não obtenha sucesso, tire dúvidas com o professor pelo email unibrattec@fredbf.com. Importantíssimo tentar começar.

1) Qual o operador bit-a-bit está sendo usado aqui? Use a representação em binário para ajudar a encontrar a resposta de cada quesito.

- a) $4 _ 1 = 2$ _____ = 4
 _____ = 1
 _____ = 2
- b) $10 _ 7 = 13$ _____ = 10
 _____ = 7
 _____ = 13
- c) $5 _ 12 = 4$ _____ = 5
 _____ = 12
 _____ = 4
- d) $7 _ 4 = 7$ _____ = 7
 _____ = 4
 _____ = 7

2) Faça uma função `crossover(n, m, pontoDeCorte)` que retorna um inteiro que representa os bits mais significativos de `n` e os bits menos significativos de `m`, de acordo com o ponto de corte, que é a posição onde o número será partido. Por exemplo:

```
main(){
    crossover(10,69,2); // retorna 13 = 0000 1101
    // 10 = 0000 1010, teremos que pegar desse n°, os bits + significativos, ou seja: 0000 1???
    // 69 = 0100 0101, teremos que pegar desse n°, os bits menos significativos, ou seja: ???? ?101
    crossover(10,69,3); // retorna 13 = 0000 0101, ou seja, metade de um + metade do outro
}
```

3) Faça a função `qtdeDeBitsParesLigados(int n)` que retorna o número de bits pares que estão ligados.

Ex:

```
main(){
    qtdeDeBitsParesLigados(73); // se 73 = 0100 1001, então qtdeDeBitsParesLigados == 2
}
```

4) Faça uma função `tresUnsConsecutivos(int n)` que retorne 1, caso o inteiro `n` possua 3 bits sequenciais com valor 1, e retorne zero caso contrário

Ex:

Ensino Superior e Técnico em Informática (UNIBRATEC)
Curso Superior de Desenvolvimento de Sistemas



Disciplina: Desenvolvimento em Hardware
Professor: Frederico Brito Fernandes



Lista de Exercícios de HW: operadores bit-a-bit e porta paralela

```
main(){
    tresUnsConsecutivos(73); // se 73 = 0100 1000, então tresUnsConsecutivos(73)== 0
    tresUnsConsecutivos(79); // se 79 = 0100 1110, então tresUnsConsecutivos(79)== 1
}
```

- 5) Faça a função `rodaPraEsquerda(int n, int nBits)` que retorna o `n` com `nBits` rotações à esquerda. Perceba que uma rotação não deve perder bits, ao contrário do operador de deslocamento. Trabalhe pensando apenas nos 8 bits para `n`.

Ex:

```
main(){
    rodaPraEsquerda(72, 2); // se 73 = 0100 1000, então rodaPraEsquerda(73,2)== 0010 0001
}
```

- 6) Escreva uma função `criptografa(int n)` que recebe um inteiro `n` com 8 bits (índices: 7,6,5,4,3,2,1,0) e que retorna esse inteiro embaralhando esses bits para a seguinte sequência (7,5,3,1,6,4,2,0)

Ex:

```
main(){
    criptografa(73); // se 73 = 0100 1001, então criptografa(73) == 0010 1001 == 41
}
```

- 7) Faça a função `descriptografa(int n)` que faz o processo invertido da questão 2

- 8) Faça uma função `pisca(int milissegundos)` que ora liga todos os bits pares e ora liga todos os bits ímpares da porta paralela (endereço `0x378`), imitando o efeito de um pisca-pisca de natal. Nesse exemplo, admita que a função `oup32(int porta,int valor)` está disponível para ser usada, e que existem 8 leds ligados no endereço `0x378`. O valor de milissegundos é usado para fazer uma espera entre a alternância, e para isso use a função `Sleep(int milissegundos)` da `windows.h`, do DevC++.

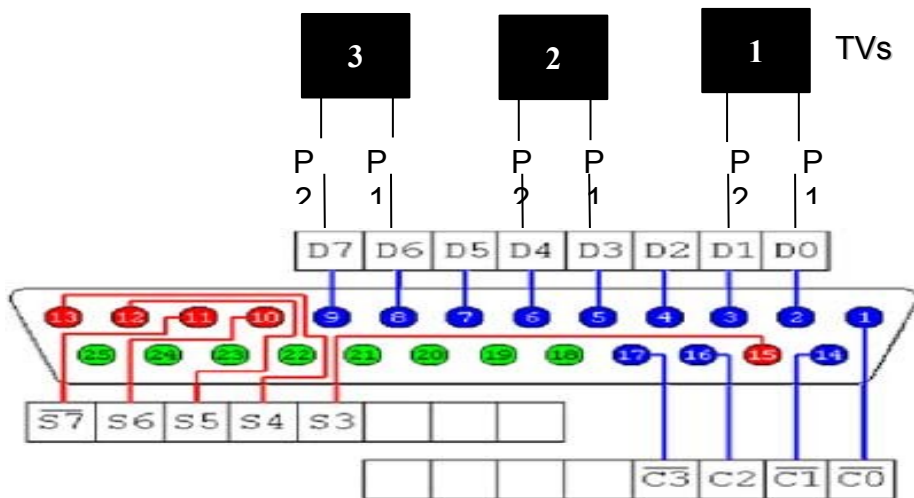
- 9) Faça uma função `piscaUmIndoEVoltando(int milissegundos)` que faz os leds pares serem ligados sequencialmente e de forma crescente, e em seguida, os leds ímpares serem ligados de forma decrescente. Admita que as funções `oup32(int porta, int valor)` e `espera (int milissegundos)` da questão anterior podem ser usadas aqui, e que os leds estão ligados da mesma maneira.

- 10) Considere que existem 3 TVs estão ligadas à porta paralela. Cada TV está ligada a dois pinos do DB25 (nomearemos de P1 e P2). A função de P1 é ligar/desligar a TV, e a função de P2 é passar para o canal seguinte. Implemente as seguintes funções:
- a) Faça uma função para ligar/desligar a TV. Se a TV está ligada, a função deve desligar a TV, ou ligá-la caso contrário. ATENÇÃO: Sua função não deve (des)ligar nenhuma outra TV, a não ser a TV passada como parâmetro.
- ```
void ligaDesligaTV(int TV)
```

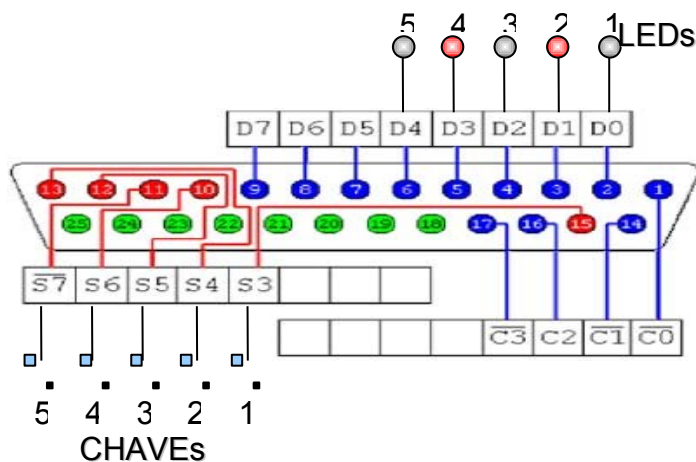
**Lista de Exercícios de HW: operadores bit-a-bit e porta paralela**

b) Faça uma função para mudar o canal da TV. Um canal é mudado quando o P2 recebe 1 e depois vai pra 0. ATENÇÃO: Sua função não deve mudar o canal de nenhuma outra TV, a não ser a TV passada como parâmetro, nem muito menos desligar outras TVs.

void mudaCanal(int TV)



11) Admita agora, que ligamos os pinos de Status em chaves, e continuamos com os LEDs ligados como no exercício anterior. Implemente em C o que se pede.



- Defina constantes para esses LEDs e constantes para as CHAVES
- Um programa principal, que fique verificando o tempo inteiro o valor da porta paralela, e que permita que, quando uma CHAVE for pressionada, seu LED correspondente será aceso