

## Lista 12 de Exercícios

### Exercícios sobre Recursividade

1. O que será impresso na tela?

```
int mdc_recurativa(int a, int b) {  
    printf("\n%d e %d", a, b);  
    if (a % b == 0) return b;  
    return mdc_recurativa (b, a % b);  
}  
  
int main() {  
    int a=32, b=18;  
    printf("\nResultado=%d",mdc_recurativa(a, b));  
}
```

2. Faça uma função recursiva para imprimir a soma dos números ímpares de um vetor.

3. Ler o texto: [www.ic.uff.br/~acopetti/progii/algoritmosFuncoesRecursivas.pdf](http://www.ic.uff.br/~acopetti/progii/algoritmosFuncoesRecursivas.pdf)

4. Faça um teste de mesa para testar 3 exemplos do programa de conversão binária.

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

```
int conversao_binaria(int n, int *vetor_binario, int tamanho, int tamanho_fixo) // o  
"tamanho_fixo" é usado somente na última chamada da função no vetor_binário
```

```
{  
    vetor_binario[tamanho] = (n % 2);          // as últimas posições do vetor receberão o  
resto dessa divisão  
    n /= 2;          // o "n" precisa ser dividido para continuar os próximos cálculos  
corretamente
```

```
    if(tamanho > 0)    // esse "if" é usado em todas as funções recursivas exceto a última  
chamada quando o tamanho for igual a zero
```

```
    {  
        tamanho--; // a função a ser chamada receberá um tamanho decrementado  
        return conversao_binaria(n, vetor_binario, tamanho, tamanho_fixo); // chama  
a função recursivamente com tamanho decrementado  
    }
```

```
    if(tamanho == 0) // apenas na última chamada será feito isso
```

```

{
    int indice, binario;    // binário será para armazenar os valores convertidos em
    binário e valor de retorno
    binario = 0;

    for(indice = 0; indice < tamanho_fixo; indice++)
    {
        if(indice == 0)
        {
            if(vetor_binario[indice] == 0) // se o vetor tem o valor zero
            {
                binario = 0;
            }
            else // se o vetor tem o valor 1
            {
                binario = 1;
            }
        }
        else
        {
            if(vetor_binario[indice] == 0) // se o valor de tal posição (indice) do
            vetor for igual a zero, então acrescenta um zero ao valor de binário
            {
                binario *= 10;
            }
            else
            {
                binario = (binario * 10) + 1; // se o valor de tal posição (indice)
            for igual a 1, então acrescenta um "1" ao valor de binário
            }
        }
    }
    return binario;
}
}

```

```

int main(void)
{
    int n; // numero a ser convertido em binário
    printf("Insira um valor inteiro decimal para ser convertido em binario: "); // pergunta
    ao usuário
    scanf("%d", &n); // leitura do numero a ser convertido em binário
    int indice, tamanho, tamanho_fixo, numero_binario;

```

```

tamanho = 0;                // zerando o tamanho do vetor

for(indice = n; indice > 0; )
{
    indice /= 2;            // laço para especificar o tamanho do vetor
    tamanho++;
}

tamanho_fixo = tamanho;      // usado na função recursiva
int vetor_binario[tamanho];  // vetor com o tamanho encontrado anteriormente no
laço
    tamanho -= 1;            // decrementa o tamanho para ser usado nessa forma na
função
    numero_binario = conversao_binaria(n, vetor_binario, tamanho, tamanho_fixo);    //
a função retornará um inteiro formatado como um "binário"
    printf("\nO numero digitado em binario e': %d", numero_binario);
    getch();
    return 0;
}

```