

ATIVIDADE DE FIXAÇÃO 1

Informações importantes:

- 1. Os exercícios que não pedirem a implementação de código favor descrever a resposta e anexar como comentário no código fonte.
- 2. Em todos os exercícios, caso entenda necessário ou que fique melhor, vocês podem fazer mais de um predicado para resolver o problema.
- 3. Para todos os exercícios vocês devem fazer um comentário no código explicando como executar o código citando exemplos.
- 4. Para todas as funções recursivas devem ser descritas o seu funcionamento.

1) Faça o rastreamento do códigos a seguir e diga o que os mesmos fazem, dizendo qual a diferença entre eles.

```
void misterio1(char b[TAM], float *dec, int i)
{
    if(i < strlen(b))
    {
        if(b[i] == '1')
            *dec = *dec + pow(2,i);
        misterio1(b,dec,++i);
    }
}
```

```
float misterio2(char b[TAM], float dec, int i)
{
    if(i < strlen(b))
    {
        dec = misterio2(b,dec,i+1);
        if(b[i] == '1')
            dec = dec + pow(2,i);
    }
    return(dec);
}
```

Obs.: A string b, contém 0s e 1s.
Obs.: Na chamada de misterio1 e de misterio2 dec e i devem ser 0
Obs.: a função pow, calcula a potência de 2 elevado a i.
Obs.: rastrear código significa, fazer uma simulação com valores mostrando o que acontece no código.

- 2) Faça uma função que dado um número inteiro devolva os divisores do mesmo, a função deve ser recursiva.
- 3) Faça uma função em C que leia um vetor de strings não ordenado e depois faça:
- (a) uma função recursiva com pendência que devolva a string de maior tamanho.
 - (b) uma função recursiva sem pendência que devolva a quantidade de strings que iniciam com vogal.
 - (c) uma função recursiva que devolva um vetor contendo somente as strings com tamanho >=4 e que iniciam com letra maiúscula.
- 4) Faça um algoritmo em C que leia 5 números inteiros e então faça:
- (a) Uma função recursiva sem pendência que calcule Mínimo Múltiplo Comum (MMC) entre os números lidos.
 - (b) Repita o exercício da letra a, mas agora faça uma função recursiva que deixa pendência.
- 5) Dado uma lista simplesmente encadeada e ordenada de pessoas:

```
struct pessoa{
    int Id;
    String Nome;
    int AnoNascimento;
    int Altura;
};
```

```
struct lista{
    Struct pessoa Info;
    Struct lista *Prox;
};
```

Faça as seguintes funções:

- (a) Uma função recursiva que devolva o nome da pessoa mais velha da lista, caso exista mais de uma devolva um vetor com os nomes.
- (b) Uma função recursiva que devolva a menor e a maior altura.
- (c) Uma função recursiva que devolva o número de pessoas com a altura mediana.
- (d) Uma função recursiva que devolva o número de pessoas abaixo de 40 anos com a altura maior do que a altura mediana.

Obs.: altura mediana = (menor altura + maior altura)/2