



UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá
Prof. Arthur Araruna
QXD0010 - Estrutura de Dados

LE1
2019.2

Nome: _____ Matrícula: _____

Observações

- Todas as respostas solicitadas nesta lista consistem de funções ou procedimentos *recursivos*.
- Perceba que para algumas questões as informações descritas no enunciado não serão suficientes para determinar todos os parâmetros.
- A menos quando pedido de outra forma, tudo o que precisar ser *calculado* ou *determinado* por uma função deverá ser fornecido pelo retorno.
- Existem dicas em algumas questões, mas elas estão com texto branco para que você possa tentar resolvê-las antes. Para ler, use a ferramenta de seleção de texto do seu visualizador.

1. Faça uma função que imprime na tela todos os números entre n e m informados (inclusive), em ordem crescente.
2. Faça uma função que imprime na tela os k primeiros números após n (inclusive), ambos informados, em ordem crescente.
3. Faça uma função que realiza a tarefa da Questão 2, porém os imprime em ordem *decrescente*.
4. Faça uma função que recebe um vetor e a quantidade de posições “preenchidas” desse vetor e retorna a soma de todos os seus elementos.
5. Faça uma função que retorna o valor do maior elemento de um vetor informado.
6. Escreva uma função para calcular o número binomial $\binom{n}{k}$ segundo a seguinte definição recursiva:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

Dica:

7. Observe cada uma das funções a seguir, que prometem retornar o maior elemento de um vetor, e determine se elas realizam a tarefa proposta. O que há de errado com as que não realizam? É possível consertá-las alterando-as pouco?

```
int maximo1(int n, int v[]) {  
    int x;  
    if (n == 1) x = v[0];  
    else {  
        x = maximo1(n-1, v);  
        if (x < v[n-1]) x = v[n-1];  
    }  
    return x;  
}
```

```
int maximo2(int n, int v[]) {  
    if (n == 1) return v[0];  
    int x = maximo2(n-1, v);  
    if (x > v[n-1]) return x;  
    else return v[n-1];  
}
```

```
int maximo3(int n, int v[]) {
    if (n == 1) return v[0];
    if (n == 2) {
        if (v[0] < v[1]) return v[1];
        else return v[0];
    }
    int x = maximo3(n-1, v);
    if (x < v[n-1]) return v[n-1];
    else return x;
}

int maximo4(int n, int v[]) {
    if (n == 1) return v[0];
    if (maximo4(n-1, v) < v[n-1])
        return v[n-1];
    else
        return maximo4(n-1, v);
}
```

8. Faça uma função que retorna o *índice* do maior elemento de um vetor informado.

Dica:

9. Diga o que a função abaixo faz. Para quais valores dos parâmetros sua resposta está correta?

```
int ttt (int x[], int n) {
    if (n == 0) return 0;
    if (x[n] > 100) return x[n] + ttt (x, n-1);
    else return ttt (x, n-1);
}
```

10. Escreva uma função que calcule o produto dos elementos *estritamente positivos* de um vetor de inteiros $V[0..n-1]$. O problema faz sentido quando todos os elementos do vetor são nulos? O problema faz sentido quando n vale 0? Quanto deve valer o produto nesses casos?
11. Podemos determinar quantos dígitos um número natural possui repetidamente dividindo-o por 10 (desconsiderando o resto) até que o resultado seja menor que 10, consistindo de um único dígito. Contamos mais 1 a cada vez que dividirmos por 10. Escreva uma função para realizar essa tarefa.
12. Escreva uma função que calcule a soma dos dígitos decimais de um natural n .
13. Escreva uma função que receba um string representando a codificação de um número natural na base binária e retorne o *valor* desse número em base decimal. Considere que o bit menos significativo é o último caractere do string. Por exemplo, se você receber '100101', o resultado deve ser 37.
14. Qual o valor de $X(4)$ se X é dada pelo seguinte código?

```
int X (int n) {
    if (n == 1 || n == 2) return n;
    else return X (n-1) + n * X (n-2);
}
```

15. Qual é o valor de $f(1, 10)$? Escreva uma função equivalente que seja mais simples.

```
double f (double x, double y) {
    if (x >= y) return (x + y)/2;
    else return f (f (x+2, y-1), f (x+1, y-2));
}
```

16. Qual o resultado da execução do seguinte programa?

```
int ff (int n) {
    if (n == 1) return 1;
    if (n % 2 == 0) return ff (n/2);
    return ff ((n-1)/2) + ff ((n+1)/2);
}
int main (void) {
    printf ("%d", ff(7));
    return 0;
}
```

17. Execute `fusc(7,0)`. Mostre a sequência, devidamente relacionadas, das chamadas a `fusc`.

```
int fusc (int n, int profund) {
    for (int i = 0; i < profund; ++i)
        printf ("  ");
    printf ("fusc(%d,%d)\n", n, profund);
    if (n == 1)
        return 1;
    if (n % 2 == 0)
        return fusc (n/2, profund+1);
    return fusc ((n-1)/2, profund+1) +
           fusc ((n+1)/2, profund+1);
}
```

18. O que acontece com a função abaixo quando passamos 25 como valor de n ? O que devemos adicionar para evitar esse problema em qualquer que seja o valor de n (considerando valores naturais)?

```
int XX (int n) {
    if (n == 0) return 0;
    else return XX (n/3+1) + n;
}
```

19. A seguinte função calcula o maior divisor comum dos inteiros estritamente positivos m e n . Escreva uma função recursiva equivalente.

```
int Euclides (int m, int n) {
    int r;
    do {
        r = m % n;
        m = n;
        n = r;
    } while (r != 0);
    return m;
}
```

20. Dadas duas strings T e P , escreva uma função que conta o número de ocorrências de P em T que não se sobreponham.
21. Escreva uma função que recebe uma string consistindo apenas de parênteses, colchetes e chaves (`'('`, `')'`, `'['`, `']'`, `'{'`, `'}'`) e retorna verdadeiro ou falso dependendo se o string recebido consiste de uma expressão com esses operadores balanceados.

Dica: