



PONTIFICIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Departamento de Ciência da Computação

PUC Minas

Disciplina Algorítmos e Estruturas de Dados I	Curso Ciência da Computação	Turno Manhã	Período 1º
Professor Felipe Cunha (felipe@pucminas.br)			

Liste X T R A – Recursão – Correção

1. Escreva um programa em C para imprimir na tela os 50 primeiros números naturais usando recursão.

SAÍDA ESPERADA:

```
Os números naturais são: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50
```

```
#include <stdio.h>

/* Função recursiva para imprimir de 1 até n */
void imprimirNaturais(int atual, int n) {
    if (atual > n) return;           // caso base: para quando ultrapassar n
    printf("%d ", atual);          // imprime o número atual
    imprimirNaturais(atual + 1, n); // chamada recursiva para o próximo
}

int main() {
    int n = 50;

    printf("Os números naturais são: ");
    imprimirNaturais(1, n);
    printf("\n");

    return 0;
}
```

2. Escreva um programa em C para calcular a soma dos números de 1 a n usando recursão.

EXEMPLO ENTRADA

Insira o último número do intervalo começando em 1: 5

SAÍDA ESPERADA:

A soma dos números de 1 a 5: 15

```
#include <stdio.h>

/* Função recursiva para somar de 1 até n */
int somaRecursiva(int n) {
    if (n == 1) return 1; // caso base
    return n + somaRecursiva(n - 1); // chamada recursiva
}

int main() {
    int n;

    printf("Insira o último número do intervalo começando em 1: ");
    scanf("%d", &n);

    int resultado = somaRecursiva(n);

    printf("A soma dos números de 1 a %d: %d\n", n, resultado);

    return 0;
}
```

3. Escreva um programa em C para imprimir a série Fibonacci usando recursão.

EXEMPLO ENTRADA

Insira o número de termos para a série (<20): 10

SAÍDA ESPERADA:

Insira o número de termos para a série (<20): 10

A série é:

1 1 2 3 5 8 13 21 34 55

```
#include <stdio.h>

/* Função recursiva que retorna o n-ésimo termo de Fibonacci */
int fibonacci(int n) {
    if (n == 1 || n == 2) return 1; // casos base
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int termos;

    printf("Insira o numero de termos para a serie (<20): ");
    scanf("%d", &termos);

    if (termos <= 0 || termos >= 20) {
        printf("Por favor, insira um valor entre 1 e 19.\n");
        return 1;
    }

    printf("A serie eh:\n");
    for (int i = 1; i <= termos; i++) {
        printf("%d ", fibonacci(i));
    }
    printf("\n");

    return 0;
}
```

4. Escreva um programa em C para imprimir os elementos do array usando recursão.

EXEMPLO ENTRADA

```
Insira o número de elementos a serem armazenados no vetor: 6
Insira 6 elementos no vetor:
elemento - 0 : 2
elemento - 1 : 4
elemento - 2 : 6
elemento - 3 : 8
elemento - 4 : 10
elemento - 5 : 12
```

SAÍDA ESPERADA:

```
Os elementos no vetor são: 2 4 6 8 10 12
```

```
#include <stdio.h>

/* Função recursiva para imprimir os elementos de um array */
void imprimirArray(int arr[], int indice, int tamanho) {
    if (indice >= tamanho) return; // caso base
    printf("%d ", arr[indice]); // imprime elemento atual
    imprimirArray(arr, indice + 1, tamanho); // chamada recursiva
}

int main() {
    int n;

    printf("Insira o número de elementos a serem armazenados no vetor: ");
    scanf("%d", &n);

    int arr[n];
    printf("Insira %d elementos no vetor:\n", n);

    for (int i = 0; i < n; i++) {
        printf("elemento - %d : ", i);
        scanf("%d", &arr[i]);
    }

    printf("Os elementos no vetor são: ");
    imprimirArray(arr, 0, n);
    printf("\n");

    return 0;
}
```

5. Escreva um programa em C para contar os dígitos de um determinado número usando recursão.

EXEMPLO ENTRADA

Entre com um número: 50

SAÍDA ESPERADA:

O número de dígitos do número é: 2

```
#include <stdio.h>

/* Função recursiva para contar dígitos */
int contaDigitos(int n) {
    if (n < 10 && n > -10) // caso base (considerando também números negativos)
        return 1;
    return 1 + contaDigitos(n / 10);
}

int main() {
    int numero;

    printf("Entre com um número: ");
    scanf("%d", &numero);

    int resultado = contaDigitos(numero);

    printf("O número de dígitos do número eh: %d\n", resultado);

    return 0;
}
```

6. Escreva um programa em C para encontrar a soma dos dígitos de um número usando recursão.

EXEMPLO ENTRADA

Insira qualquer número para encontrar a soma dos dígitos: 25

SAÍDA ESPERADA:

A soma dos dígitos de 25 = 7

```
#include <stdio.h>

/* Função recursiva para somar os dígitos de um número */
int somaDigitos(int n) {
    if (n == 0) return 0; // caso base
    return (n % 10) + somaDigitos(n / 10); // pega o último dígito + recursão
}

int main() {
    int numero;

    printf("Insira qualquer número para encontrar a soma dos dígitos: ");
    scanf("%d", &numero);

    if (numero < 0) numero = -numero; // tratar números negativos

    int resultado = somaDigitos(numero);

    printf("A soma dos dígitos de %d = %d\n", numero, resultado);

    return 0;
}
```

7. Escreva um programa em C para encontrar MDC de dois números usando recursão.

EXEMPLO ENTRADA

Insira o Primeiro número: 10
Segundo número de entrada: 50

SAÍDA ESPERADA:

O MDC de 10 e 50 é: 10

```
#include <stdio.h>

/* Função recursiva para calcular MDC usando o algoritmo de Euclides */
int mdc(int a, int b) {
    if (b == 0)
        return a;
    return mdc(b, a % b);
}

int main() {
    int num1, num2;

    printf("Insira o Primeiro numero:");
    scanf("%d", &num1);

    printf("Segundo numero de entrada:");
    scanf("%d", &num2);

    int resultado = mdc(num1, num2);

    printf("O MDC de %d e %d eh: %d\n", num1, num2, resultado);

    return 0;
}
```

8. Escreva um programa em C para obter o maior elemento de um vetor usando recursão.

EXEMPLO ENTRADA

Insira o número de elementos a serem armazenados no vetor: 5

Insira 5 elementos no vetor:

elemento - 0 : 5
elemento - 1 : 10
elemento - 2 : 15
elemento - 3 : 20
elemento - 4 : 25

SAÍDA ESPERADA:

O maior elemento de uma matriz é: 25

```
#include <stdio.h>

/* Função recursiva para encontrar o maior elemento do vetor */
int maiorElemento(int arr[], int n) {
    if (n == 1) {
        return arr[0]; // caso base
    }

    int maiorAnterior = maiorElemento(arr, n - 1);
    return (arr[n - 1] > maiorAnterior) ? arr[n - 1] : maiorAnterior;
}

int main() {
    int n;

    printf("Insira o numero de elementos a serem armazenados no vetor: ");
    scanf("%d", &n);

    int arr[n];
    printf("Insira %d elementos no vetor:\n", n);

    for (int i = 0; i < n; i++) {
        printf("elemento - %d : ", i);
        scanf("%d", &arr[i]);
    }

    int maior = maiorElemento(arr, n);

    printf("O maior elemento de uma matriz eh: %d\n", maior);

    return 0;
}
```

9. Escreva um programa em C para reverter uma string usando recursão.

EXEMPLO ENTRADA

String de entrada: w3resource

SAÍDA ESPERADA:

A string reversa é: ecruos3w

```
#include <stdio.h>

/* Função recursiva para imprimir a string invertida */
void reverterString(char str[], int indice) {
    if (str[indice] == '\0') {
        return; // caso base: fim da string
    }
    reverterString(str, indice + 1); // chama recursivamente
    printf("%c", str[indice]); // imprime ao "voltar" da recursão
}

int main() {
    char str[100];

    printf("String de entrada: ");
    scanf("%s", str);

    printf("A string reversa eh: ");
    reverterString(str, 0);
    printf("\n");

    return 0;
}
```

10. Escreva um programa em C para imprimir números pares ou ímpares em um determinado intervalo usando recursão.

EXEMPLO ENTRADA

Insira o intervalo de impressão a partir de 1: 10

SAÍDA ESPERADA:

Todos os números pares de 1 a 10 são: 2 4 6 8 10

Todos os números ímpares de 1 a 10 são: 1 3 5 7 9

```
#include <stdio.h>

/* Função recursiva para imprimir pares */
void imprimirPares(int atual, int fim) {
    if (atual > fim) return; // caso base
    if (atual % 2 == 0) printf("%d ", atual);
    imprimirPares(atual + 1, fim); // chamada recursiva
}

/* Função recursiva para imprimir ímpares */
void imprimirImpares(int atual, int fim) {
    if (atual > fim) return; // caso base
    if (atual % 2 != 0) printf("%d ", atual);
    imprimirImpares(atual + 1, fim); // chamada recursiva
}

int main() {
    int n;

    printf("Insira o intervalo de impressão a partir de 1: ");
    scanf("%d", &n);

    printf("Todos os números pares de 1 a %d são: ", n);
    imprimirPares(1, n);
    printf("\n");

    printf("Todos os números ímpares de 1 a %d são: ", n);
    imprimirImpares(1, n);
    printf("\n");
}

return 0;
}
```

11. Escreva um programa em C para verificar se uma determinada String é Palíndromo ou não.

EXEMPLO ENTRADA

Insira uma palavra para verificar o palíndromo: mom

SAÍDA ESPERADA:

A palavra inserida é um palíndromo.

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

/* Função recursiva para verificar se é palíndromo */
int ehPalindromo(char str[], int inicio, int fim) {
    if (inicio >= fim) return 1; // caso base: chegou ao meio
    if (str[inicio] != str[fim]) return 0; // caracteres diferentes? não é palíndromo
    return ehPalindromo(str, inicio + 1, fim - 1); // chamada recursiva
}

int main() {
    char str[100];

    printf("Insira uma palavra para verificar se é palíndromo: ");
    scanf("%s", str);

    int tamanho = strlen(str);

    if (ehPalindromo(str, 0, tamanho - 1))
        printf("A palavra inserida é um palíndromo.\n");
    else
        printf("A palavra inserida não é um palíndromo.\n");

    return 0;
}
```

12. Escreva um programa em C para encontrar a primeira letra maiúscula em uma string usando recursão.

EXEMPLO ENTRADA

Insira uma string para incluir uma ou mais letras maiúsculas:
testString

SAÍDA ESPERADA:

A primeira letra maiúscula que aparece na string testString é S.

```
#include <stdio.h>
#include <ctype.h> // para isupper

/* Função recursiva para encontrar a primeira letra maiúscula */
char primeiraMaiuscula(char str[], int i) {
    if (str[i] == '\0') {
        return '\0'; // caso base: fim da string sem encontrar maiúscula
    }
    if (isupper(str[i])) {
        return str[i]; // encontrou a primeira maiúscula
    }
    return primeiraMaiuscula(str, i + 1); // chamada recursiva
}

int main() {
    char str[100];

    printf("Insira uma string para incluir uma ou mais letras maiúsculas: ");
    scanf("%s", str);

    char resultado = primeiraMaiuscula(str, 0);

    if (resultado != '\0') {
        printf("A primeira letra maiuscula que aparece na string %s eh %c.\n", str, resultado);
    } else {
        printf("Nao ha letras maiusculas na string %s.\n", str);
    }

    return 0;
}
```