SENAI CIMATEC

SISTEMA FIEB

TECNOLOGIA E INOVAÇÃO EM PROL DA INDÚSTRIA

SENAI CIMATEC

SISTEMA FIEB

Programador web - SENAI

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Programação de Aplicativo - 140h

Profª: Francisleide Almeida

Recursividade em C

- Chamamos de recursividade ou recursão quando uma função chama a si mesma.
- Em uma função recursiva, a cada chamada é criada na memória uma nova ocorrência da função com comandos e variáveis “isolados” das ocorrências anteriores.

Recursividade em C

- A função é executada até que todas as ocorrências tenham sido resolvidas.
- Problema: como fazê-la parar?
- É fácil cair num loop infinito recursivo o qual pode inclusive esgotar a memória.

Recursividade em C

- **Caso base**: é o caso mais simples. É usada uma condição em que se resolve o problema com facilidade.
- **Chamadas Recursivas**: procuram simplificar o problema de tal forma que convergem para o caso base.

Recursividade em C

- Vantagens da recursividade:
  - Torna a escrita do código mais simples e elegante, tornando-o fácil de entender e de manter.
- Desvantagens da recursividade
  - Quando o loop recursivo é muito grande é consumida muita memória nas chamadas a diversos níveis de recursão, pois cada chamada recursiva aloca memória para os parâmetros, variáveis locais e de controle.
  - Em muitos casos uma solução iterativa gasta menos memória, e torna-se mais eficiente em termos de performance do que usar recursão.

Recursividade em C

- Exemplo 1:
  - Vamos criar uma função soma(int n).
  - Se n=5, essa função deve retornar: soma(5) = 5 + 4 + 3 + 2 + 1
  - Se n=4, essa função deve retornar: soma(4) = 4 + 3 + 2 + 1
  - Se n=3, essa função deve retornar: soma(3) = 3 + 2 + 1
  - …
  - **Veja que:**
  - soma(5) = 5 + 4 + 3 + 2 +1 = 5 + soma(4)
  - O mesmo para:
  - soma(4) = 4 + 3 + 2 + 1 = 4 + soma(3)
  - Ou seja, temos a fórmula geral:
  - soma(n) = n + soma(n-1)

  - Ou seja:
  - soma(n) = n + soma(n-1) = n + (n-1) + soma(n-2) = n + (n-1) + (n-2) + soma(n-3)...

Recursividade em C

- Exemplo 1:
  - E onde essa soma para? Para quando o último elemento dessa soma for 1.
  - Então:
  - soma(n) = n +(n-1) + (n-2) + (n-3) + ... + 1
  - Agora vamos traduzir essa fórmula em termos de programação.
  - A função recebe um número, e a primeira coisa que ela deve fazer é ver se esse valor é 1.
  - Se for, deve retornar 1, afinal:
  - soma(1) = 1
  - E se não for 1, deve retornar:
  - n + soma(n-1)

Recursividade em C

- Exemplo 1:

```c
#include <stdio.h>

int soma(int n)
{
    if(n == 1)
        return 1;
    else
        return ( n + soma(n-1) );
}

int main()
{
    int n;
    printf("Digite um inteiro positivo: ");
    scanf("%d", &n);

    printf("Soma: %d\n", soma(n));
}
```

Recursividade em C

- Exemplo clássico de recursividade: **fatorial**.

```c
//Cálculo de fatorial com função recursiva
#include <stdio.h>
#include <conio.h>

//protótipo da função fatorial
double fatorial(int n);

int main(void)
{
  int numero;
  double f;

  printf("Digite o numero que deseja calcular o fatorial: ");
  scanf("%d",&numero);

  //chamada da função fatorial
  f = fatorial(numero);

  printf("Fatorial de %d = %.0lf",numero,f);

  getch();
  return 0;
}

//Função recursiva que calcula o fatorial
//de um numero inteiro n
double fatorial(int n)
{

  double vfat;

  if ( n <= 1 )
     //Caso base: fatorial de n <= 1 retorna 1
     return (1);
  else
  {
     //Chamada recursiva
     vfat = n * fatorial(n - 1);
     return (vfat);
  }
}
```

Atividade

1 - Faça uma função recursiva que calcule e retorne o N - ésimo termo da sequência Fibonacci. Alguns números desta sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

2 - Faça uma função recursiva que permita inverter um número inteiro N. Ex: 123 – 321

3 - Faça uma função recursiva que permita somar os elementos de um vetor de inteiros.

4 - Crie um programa em C, que contenha uma função recursiva que receba dois inteiros positivos k e n e calcule k^n Utilize apenas multiplicações. O programa principal deve solicitar ao usuário os valores de k e n e imprimir o resultado da chamada da função.

5 - Os números tetranacci iniciam com quatro termos pré-determinados e a partir daí todos os demais números são obtidos pela soma dos quatro números anteriores. Os primeiros números tetranacci são: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208...Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tetranacci.