Centro de Matemática, Computação e Cognição – UFABC
MCTA028-15 – Programação Estruturada – 3º Quadrimestre de 2023
http://professor.ufabc.edu.br/~m.sambinelli/courses/2023Q3-PE/index.html

Laboratório 5: recursão

Atenção: de agora em diante está terminantemente **proibido** usar arrays estáticos de tamanho variado como os exibidos abaixo:

```
int n;
int vetor[n]; // nunca definir a dimensão de um array com uma variável
```

Essa proibição aplica-se a tudo: listas, trabalho e avaliação. Qualquer programa usando o tipo de construção acima receberá nota **zero**. Todo array dinâmico deverá ser construído usando alocação dinâmica de memória:

```
int* vetor = calloc(n, sizeof(int)); // ou
int* vetor2 = malloc(n * sizeof(int));
```

Instruções

- Em todos os seus programas você deve gerenciar corretamente a memória, liberando toda a memória requerida pelo seu programa após o término do uso. Programas com vazamento de memória receberão uma penalização de 25% do valor da nota total do exercício. Você pode verificar se o seu programa possui vazamento de memória com o comando

  ```
  valgrind --leak-check=full /caminho/para/o/seu/programa
  ```

  Para que o comando acima funcione, você deve habilitar a *flag* de *debug* do seu compilador. O exemplo a seguir ilustra como deve ser feito caso você use o *gcc* para compilar o seu programa

  ```
  gcc -Wall -Wextra -Wvla -g -std=c99 arquivo.c
  ```

- Em vários exercícios, eu peço a vocês para que escrevam uma função de um determinado tipo. Além de escrever essa função, vocês também devem escrever uma função `main()` que irá usar essa função com dados fornecidos pelo usuário. Ou seja, a sua `main()` deverá pedir a entrada para o usuário e passar esses dados como parâmetro para a função que você desenvolveu. Requisite esses dados imprimindo mensagens na tela, para que o professor saiba o que digitar quando estiver corrigindo o seu trabalho.

Questão 1. Implemente a função recursiva

```
void count_down(int n);
```

Essa é uma função recursiva que faz uma contagem regressiva. Um exemplo da saída impressa por essa função quando chamamos `count_down(5)` poderia ser:

```
5
4
3
2
1
```

```
ACABOU!
```

Questão 2. Implemente a função recursiva

```
void count_up(unsigned int n);
```

Nesse exercício, você **não** pode usar nenhuma variável global e nem mudar a assinatura da função. Essa é uma função recursiva que faz uma contagem progressiva. Um exemplo da saída impressa por essa função quando chamamos `count_up(5)` poderia ser:

```
0
1
2
3
4
5
```

Questão 3. Implemente a função recursiva

```
int soma(int v[], int n);
```

Essa função recursiva retorna a soma de todos os elementos do vetor `v`.

Questão 4. Implemente a função recursiva chamada `inverte`. Um dos parâmetros que essa função irá receber é `char word[]`. Você pode passar outros parâmetros se sentir necessidade. Como resultado, essa função deve inverter a palavra recebida, i.e., se a `word` armazenar a palavra "abobora", após a execução da função, ela armazenará a palavra "aroboba".

Questão 5. Adapte o programa que resolve o problema da Torre de Hanoi visto em sala de aula. Ao invés de exibir os movimentos, imprima o número de movimentos necessários para mover os n discos.

Questão 6. Implemente uma função recursiva para gerar a sequência de Collatz para um número inteiro positivo dado. A função deve seguir as regras da sequência de Collatz:

- Se o número atual for par, divida-o por 2.
- Se o número atual for ímpar, multiplique-o por 3 e some 1.

A função deve imprimir cada termo da sequência até que o número atual seja 1, momento em que a recursão deve parar.

Como exemplo, temos que a sequência de Collatz para o número 7 é:

```
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Questão 7. Escreva uma função recursiva que calcule $\binom{n}{k}$, para $n \geq 0$ e $k \geq 0$. Para isso, observe a seguinte propriedade:

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

Questão 8. Escreva um programa recursivo que, dado um número positivo n fornecido pelo usuário, seja capaz de imprimir todas as combinações dos n primeiros naturais. Para $n = 3$, um exemplo de saída seria:

```
1
2
3
12
13
23
123
```

Questão 9. Escreva um programa recursivo que, dados n números positivos e um inteiro k, onde $1 \leq k \leq n$, seja capaz de imprimir todas as combinações de k elementos dos valores fornecidos. Um exemplo de entrada possível seria

```
4 3
1
7
2
8
```

Na entrada acima, o usuário informa que entrará com 4 valores e que quer uma combinação de 3 em 3. Uma possível saída seria:

```
1 7 2
1 7 8
1 2 8
7 2 8
```