

Curso: Desenvolvimento Web e Multimédia / Aplicações Móveis

U.C.: Algoritmos e Estruturas de Dados I

Folha de Exercícios FE05

Docente	Marta Martinho / Patrícia Leite
Tema	Recursividade e apontadores
Turma	DWM/AM
Ano Letivo	2020/2021 - 1º Semestre
Duração da Aula	2 horas

Enunciado

1. Crie uma função iterativa que calcule o resultado do fatorial de um número ($5! = 5 \times 4 \times 3 \times 2 \times 1$).
2. Implemente a função anterior de forma recursiva.
3. Implemente uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente.
4. Implemente uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem decrescente.
5. Implemente uma função recursiva que receba um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem crescente.
6. Faça uma função recursiva que receba um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem decrescente.
7. Crie uma função iterativa que receba um intervalo de números inteiros positivos e que calcule o resultado do somatório dos números no intervalo dado.
8. Implemente a função anterior de forma recursiva.
9. Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.
10. Crie uma função iterativa que calcule a potência matemática de um dado número.
11. Implemente a função anterior de forma recursiva.
12. O máximo divisor comum dos inteiros x e y é o maior inteiro que é divisível por x e y. Escreva uma função recursiva mdc em C, que retorna o máximo divisor comum de x e y.

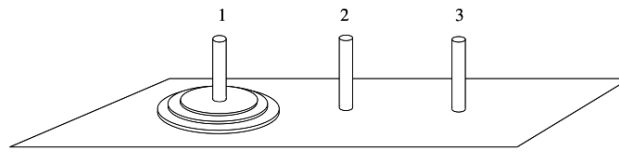
y. O mdc de x e y é definido como segue: se y é igual a 0, então $\text{mdc}(x,y)$ é x; caso contrário, $\text{mdc}(x,y)$ é $\text{mdc}(y, x\%y)$, onde % é o operador resto.

13. Implemente 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...

14. Implemente uma função recursiva que permita inverter um número inteiro N. Ex: 123 - 321

15. Implemente uma função recursiva que determine quantas vezes um dígito K ocorre em um número natural N. Por exemplo, o dígito 2 ocorre 3 vezes em 762021192.

16. O quebra-cabeças Torres de Hanoi é muito antigo e conhecido, sendo constituído de um conjunto de N discos de tamanhos diferentes e três pinos verticais, nos quais os discos podem ser encaixados.



Cada pino pode conter uma pilha com qualquer número de discos, desde que cada disco não seja colocado acima de outro disco de menor tamanho.

A configuração inicial consiste em ter todos os discos no pino 1. O objetivo do quebra-cabeças é mover todos os discos para um dos outros pinos, sempre obedecendo à restrição de não colocar um disco sobre outro menor.

Desenvolva um programa que determine quantos movimentos de trocar um disco de um pino para outro serão executados pelo algoritmo acima para resolver o quebra-cabeça.

17. Analise o seguinte bloco de código:

```
int x, y, *p;
y = 0;
p = &y;
x = *p;
x = 4;
(*p)++;
--x;
(*p) += x;
printf("x=%d y=%d *p=%d", x, y, *p);
```

Indique qual o *output* do programa e de seguida implemente-o de forma a confirmar o seu raciocínio.

18.Desenvolva uma aplicação que permita a troca de valores armazenados em duas variáveis, ou seja colocar o valor da variável *x* na variável *y* e vice versa. Implemente o procedimento `trocaValores(x, y)` para resolver o problema.

19.Desenvolva o procedimento `verificaPrimo(numero, primo)` que verifica se um número é primo ou não.