

Leia com atenção o enunciado apresentado por cada exercício, e desenvolva o algoritmo solicitado.

- 1) Faça uma função recursiva para elevar um valor base a potência do valor expoente.
- 2) Faça uma função recursiva que calcule o M.D.C. (máximo divisor comum) entre dois valores fornecidos pelo usuário. Por exemplo: m.d.c. de 12 e 20 é 4.

| | |
|--------|-------------|
| 12, 20 | 20 - 12 = 8 |
| 12, 8 | 12 - 8 = 4 |
| 4, 8 | 8 - 4 = 4 |
| 4, 4 | M.D.C. |

- 3) Faça uma função recursiva que calcule o M.D.C. pelo algoritmo de Euclides. Por exemplo, divide-se a por b e obtêm-se o quociente q e o resto r. Se r for zero, b é o m.d.c; se não for, divide-se b por r e assim sucessivamente até encontrar um resto zero. O último divisor é o M.D.C.
- 4) Faça um programa que receba um valor n, e imprima a contagem regressiva a partir deste valor. Por exemplo, se o usuário digitar 5, o programa irá imprimir 5, 4, 3, 2, 1, 0.
- 5) Faça uma função recursiva que recebe um vetor preenchido e a quantidade de posições deste vetor, e retorna a soma de todos os elementos do vetor.
- 6) Faça uma função recursiva que realize a multiplicação entre dois valores int, sem utilizar o operador *.
- 7) Faça uma função recursiva que receba os valores de X e N e realize o cálculo da seguinte progressão geométrica:

$$1 + x + x^2 + x^3 + x^4 + \dots + x^n$$

- 8) Faça o teste de mesa do algoritmo recursivo abaixo (procure fazer a mão, e não utilizando o computador), e responda a pergunta:

```
1  #include <stdio.h>
2
3  int XXX (int n, int m) {
4      if ((n == m) || (m == 0)) {
5          return 1;
6      }
7      return (XXX(n-1,m) + XXX(n-1, m-1));
8  }
9
10 int main() {
11     int n = 5, m = 3;
12     printf("%d", XXX(n,m));
13     return 0;
14 }
```

Qual o valor de x (5,3) ?

9) Faça o teste de mesa do programa abaixo (procure fazer a mão, e não utilizando o computador), e informe o resultado final.

```
1  #include <stdio.h>
2
3  int qualquer (int n){
4      if(n<=2){
5          return n;
6      }
7      return (qualquer(n-1) + qualquer(n-2) + qualquer(n-3));
8  }
9
10 int main() {
11     int n = 6;
12     printf("%d", qualquer(n));
13     return 0;
14 }
```

10) Escreva um programa que leia quatro valores inteiros positivos n_A , n_B , t_A e t_B , representando respectivamente as populações atuais de dois países A e B e as taxas de crescimento anual dessas populações, e determine se o país menos populoso poderá ultrapassar a população do outro país, supondo que as taxas de crescimento dessas populações não variam. Em caso afirmativo, o programa deverá determinar também o número de anos necessários para que isto aconteça. Utilize funções recursivas para resolver o problema.

11) Escreva uma função recursiva, `int SomaSerie (int i, int j, int k)`, que imprime na tela a soma de valores do intervalo $[i,j]$, com incremento k .

12) Faça um programa recursivo que contenha uma função para calcular e retornar o resultado da seguinte série:

$$\frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \frac{4}{N-3} + \dots$$

N é um valor inteiro maior ou igual a 1, digitado pelo usuário. A série deve ser calculada até que o denominador seja igual a 1. O valor de N deverá ser fornecido pelo usuário.

13) Faça um programa de busca binária recursivo. Crie um vetor de 15 posições e preencha-o utilizando `rand`, e cuidando para não haver valores repetidos. A seguir ordene-o (tanto o preenchimento quanto a ordenação não precisam ser recursivos) e solicite ao usuário que entre com o valor que deseja procurar no vetor. Esse valor será levado a uma função de busca binária recursiva, que irá retornar verdadeiro se o elemento existir no vetor e falso se não existir. Exiba esta mensagem na tela.

14) Faça uma função recursiva, em linguagem C, que calcule o valor da série S descrita a seguir para um valor $n > 0$ a ser fornecido como parâmetro para a mesma.

$$S = 2 + \frac{5}{2} + \frac{10}{3} + \frac{17}{4} + \dots + \frac{1 + n^2}{n}$$

15) Pode-se calcular o resto da divisão, MOD, de x por y, dois números inteiros, usando-se a seguinte definição:

$$MOD(x, y) = \begin{cases} MOD(|x| - |y|), & \text{se } |x| > |y| \\ |x|, & \text{se } |x| < |y| \\ 0 & \text{se } |x| = |y| \end{cases}$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. A função deve retornar -1 caso não seja possível realizar o cálculo. Além disso, crie um algoritmo que leia os dois valores inteiros e utilize a função criada para calcular o quociente de x por y, e imprima o valor computado.

16) Pode-se calcular o quociente da divisão, DIV, de x por y, dois números inteiros, usando-se a seguinte definição:

$$DIV(x, y) = \begin{cases} 1 + DIV(|x| - |y|, |y|), & \text{se } |x| > |y| \\ 0, & \text{se } |x| < |y| \\ 1, & \text{se } |x| = |y| \end{cases}$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. A função deve retornar -1 caso não seja possível realizar o cálculo. Além disso, crie um algoritmo que leia os dois valores inteiros e utilize a função criada para calcular o quociente de x por y, e imprima o valor computado.

17) Um problema típico em ciência da computação consiste em converter um número da sua forma decimal para a forma binária. Por exemplo, o número 12 tem a sua representação binária igual a 1100. A forma mais simples de fazer isso é dividir o número sucessivamente por 2, onde o resto da i-ésima divisão vai ser o dígito i do número binário (da direita para a esquerda).

Por exemplo:

12 / 2 = 6, resto 0 (1º dígito da direita para esquerda)
6 / 2 = 3, resto 0 (2º dígito da direita para esquerda)
3 / 2 = 1 resto 1 (3º dígito da direita para esquerda)
1 / 2 = 0 resto 1 (4º dígito da direita para esquerda).
Resultado: 12 = 1100.

Faça uma função recursiva que dado um número decimal imprima a sua representação binária corretamente.