EXERCÍCIOS DE FUNÇÕES

1. Criar uma função para somar dois números e retornar o resultado da soma. A assinatura da função é: int somar(int a, int b)
   Criar um main para fazer a chamada a esta função e imprimir o resultado retornado por ela.
2. Seguindo a mesma ideia do exercício 1, implementar outras 3 funções para: subtrair, multiplicar e dividir dois valores, retornando o resultado da operação e fazendo o main imprimir este resultado.
3. Escreva uma função que receba um número inteiro como parâmetro (que corresponde ao ano) e retorne 1 caso ele seja bissexto e 0 caso contrário.
4. Escreva uma função com três números inteiros como parâmetros de entrada: **dia, mês e ano.** Estes valores representam uma data. A função deve **calcular e imprimir o dia seguinte da data passada como parâmetro**. No main, ler os valores do teclado e utilizar esta função. Utilize a função do exercício anterior para saber se o ano é ou não bissexto.
5. Criar uma função que receba um parâmetro inteiro (n) e retorne a soma: n + n -1. Caso n seja zero, a função deve retornar zero. Criar um main para ler um valor inteiro n da entrada padrão e somar os valores retornados pela função para cada valor de 0 até n.
6. Alterar a função do exercício anterior para retornar: n - 1. Não se esqueça de manter o tratamento do zero. Executar o main novamente e verificar o resultado.
7. Criar uma função que retorne o valor para o seguinte cálculo: x^y. Criar um main para ler os valores de x e y e imprimir o resultado da operação. Não é permitido usar a função pow(...).
8. Criar uma função que imprima os números de primos 1 a n (máximo 1000). Criar um main para ler o número e executar a função.
9. Criar uma função que receba dois parâmetros inteiros e retorne o MDC (máximo divisor comum) entre os números.
10. Considere um número N sendo quadrado perfeito. A raiz quadrada deste número pode ser calculada da seguinte forma: calcular a soma dos números ímpares consecutivos cuja soma seja igual a N. A quantidade de elementos utilizados nesta soma é o valor da raiz quadrada de N. Baseado nestas informações, crie uma função que receba um número inteiro N e retorne se o número é ou não um quadrado perfeito. Criar uma função main para ler um valor do teclado e imprimir todos os quadrados perfeitos de 1 até N. Não é permitido usar a função sqrt(...).

EXERCÍCIOS DE PONTEIROS

1. Implemente uma função para contar a quantidade de caracteres de uma string. A assinatura da função deve ser: int tamanho(char * pMsg)
2. Implemente uma função para imprimir caractere por caractere de de uma string. A assinatura da função deve ser: void imprimir(char * pMsg)
3. Implemente uma função para concatenar duas strings de caracteres. A assinatura da função deve ser: void concatenar(char * destino, char * origem)

Implementar um main para criar as strings e chamar as funções acima, imprimindo o conteúdo das strings antes e após as chamadas das funções e o valor retornado pela função `tamanho`.

1. Implemente uma função que copia um vetor de caracteres para outro vetor de forma invertida. A assinatura da função deve ser: void inverter(char * str1, char * str2).
2. Implemente uma função para verificar se o conteúdo de uma string (str2) está presente em qualquer posição de uma outra string (str1). A assinatura da função deve ser: int buscarSubString(char * str1, char * str2). Obs: não utilizar os índices dos vetores. Usar apenas os ponteiros. Não é permitido usar a função para achar o tamanho das strings.
3. Implemente uma função para recortar uma string. O recorte deve ser feito de acordo com a posição inicial e final indicada. Uma nova string deve ser retornada. A assinatura da função deve ser: char * subString(char * str, int inicio, int fim).

Implemente as seguintes funções:

1. **char* strchr(char *s, char ch);** que retorna o endereço da última ocorrência de **ch** em **s;** caso não exista, retorna NULL. (Note que é o endereço, e não o índice).
2. **char* strstr(char *str1, char *str2)**; Retorna o endereço de str1 em que ocorre pela primeira vez a substring str2. Caso não exista, retorna NULL.
3. **char* First_Vogal(char *s);** Retorna o endereço em que ocorre a primeira vogal na string s. caso não exista, retorna NULL.
4. **char* strins(char *dest, char *orig);** Insere a string orig no início da string dest, retornando dest.

   Exemplo:

   char s[100] = “Autonoma”;
   strins(s,”Universidade”);
   printf(s); → UniversidadeAutonoma