

## Lista de exercícios 2

### Funções

1. Escreva uma função que receba um número `double` como argumento e retorne, em outros dois argumentos passados por referência, a parte inteira e a parte decimal desse número. Não utilize a função `modf` declarada no arquivo de inclusão `cmath`.
2. Escreva uma função recursiva chamada `potencia` que aceite dois argumentos inteiros positivos `i` e `j`. A função retorna o `i` elevado à potência `j`.
3. Um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo, o número 6 é um número perfeito, pois:  $6 = 1 + 2 + 3$ . O próximo número perfeito é o 28, pois:  $28 = 1 + 2 + 4 + 7 + 14$ . Escreva uma função que determina se um determinado número é ou não perfeito. Use essa função em um programa que imprima todos os números perfeitos entre 1 e 1000.
4. Escreva uma função que retorne o máximo divisor comum (mdc) entre dois inteiros. O máximo divisor comum entre dois inteiros é o maior número que é fator comum dos dois inteiros. Por exemplo, os divisores comuns de 12 e 18 são 1,2,3 e 6, logo o máximo divisor comum entre 12 e 18 é 6.

### Arrays

5. Escreva uma função de protótipo `void strinv(char s[])` que inverta os caracteres de uma string. Por exemplo, se a string for "ABCDEF", deve ser convertida para "FEDCBA". Use somente o array `s`, não crie um novo array para resolver o problema.
6. Um inteiro primo é qualquer inteiro que só pode ser dividido exatamente por si mesmo e por 1. O crivo de Eratóstenes é um método para se encontrar números primos. Para encontrar todos os números primos menores ou iguais a um inteiro  $n$  basta realizar o seguinte procedimento:
  - a. Crie uma lista de inteiros consecutivos de 2 até  $n$
  - b. Marque todos os elementos da lista inicialmente como primos
  - c. Inicialmente, faça  $p=2$ , o primeiro número primo
  - d. Marque como não primo todos os números superiores a  $p$  que são múltiplos de  $p$
  - e. Encontre o próximo número que não está marcado como não primo da lista e faça  $p$  igual a esse número
  - f. Repita os passos *d* e *e* até que  $p^2$  seja maior que  $n$

Escreva um programa em C++ que utilize array e que implementa o procedimento descrito acima.

7. Escreva um programa que lê uma linha de texto e imprime a linha com todos os números substituídos por 'x'. Use a função `isdigit` declarada no arquivo de inclusão `cctype`. Por exemplo:

Input: Meu id é john79 e minha senha de 4 dígitos é 1846

Output: Meu id é johnxx e minha senha de x dígitos é xxxx`

A função `isdigit` tem o seguinte protótipo: `int isdigit(int c)`, onde `c` é o caractere a ser verificado, convertido para inteiro. A função retorna valor zero se `c` não é um dígito decimal e valor diferente de zero se `c` é um dígito decimal.

### Ponteiros

8. Escreva uma função que inverta a ordem dos caracteres de uma cadeia de caracteres que ela recebe com argumento. O protótipo da função deve ser: `void inverte_string(char* pstring)`.
9. Um problema com arrays dinâmicos é que uma vez que o array é criado usando o operador `new` seu tamanho não pode ser alterado. Por exemplo, você pode querer adicionar ou remover entradas do array. Crie um programa que cria um array dinâmico de 5 inteiros. A seguir, escreva duas funções
- `int* adiciona_entrada(int * v, int& tamanho, int novo)`: Essa função deve criar um novo array dinâmico um elemento maior que `v`, copiar todos os elementos de `v` para o novo array, incrementar `tamanho` e retornar o array criado
  - `int * apaga_entrada(int * v, int& tamanho, int apagar)`: Essa função deve procurar o valor `apagar` em `v`. Se não encontrar, a solicitação deve ser ignorada e `v` deve ser retornado sem modificações. Se forem encontradas `n` entradas em `v` iguais a `elemento_apagar`, crie um novo array `n` elementos menor que `v`, libere a memória ocupada por `v`, diminua `tamanho` em `n` unidades e retorne o novo array.

### Métodos numéricos

10. Escreva um programa que calcula a taxa interna de retorno dos fluxos de caixa descritos na tabela abaixo. Utilize precisão de 0.001%. Resolva o problema utilizando dois métodos: Bisseção e Newton-Raphson.

Tempo (anos)	Valor do fluxo de caixa
0,0	-1000,00
0,5	100,00
1,0	200,00
1,5	100,00
2,0	200,00

2,5	100,00
3,0	200,00
3,5	100,00
4,0	200,00
4,5	100,00
5,0	200,00

A taxa interna  $r$  de um conjunto de  $n$  fluxos de caixa é aquela que satisfaz a equação:

$$0 = f(r) = \sum_{k=1}^n \frac{C_k}{(1+r)^{t_k}}$$