

Lista 3 (EXERCÍCIOS)

NOTA: Para os exercícios desta lista e das seguintes, utilize as bibliotecas do C++ e não as da linguagem C, quando possível. Exemplo: a) utilize `cin` no lugar de `scanf`; b) utilize `cout` no lugar de `printf`; etc.

1) Escreva um programa que lê um nome (string) e uma idade (int). Imprima os dados lidos.

2) Repita o programa anterior, desta vez armazenando os pares de dados em um vetor (vector). Imprima o conteúdo do vetor após a leitura.

3) Inicialize um vetor de inteiros e atribua os elementos 5, 9, -1, 200 e 0. Imprima o vetor. Ordene o vetor e imprima novamente.

4) Repita o programa anterior, desta vez utilizando strings. Atribua os elementos “Linus”, “Stroustrup”, “Godel”, “Tanenbaum”, “Turing” e “Dijkstra”.

5) Escreva um programa que lê pares de nomes e idades de um arquivo texto. Ordene os dados por nomes e imprima. Ordene os dados por idade e imprima.

6) Continue o programa anterior, desta vez salvando os dados ordenados em dois arquivos textos. Um para nomes e outro para idades.

7) Escreva um programa que recebe um arquivo, abre-o em modo binário e verifica se é um executável do Windows ou do Linux.

Dica: procure pelo identificador (*magic number*) utilizado nos executáveis.

8) Faça um programa que imprima a tabela ASCII, contendo caracteres e seus valores em decimal, octal e hexadecimal. Pesquise e utilize os modificadores de `cin`.

9) Escreva um programa que receba um valor numérico do usuário e o imprime. O programa deverá apresentar este comportamento até que o usuário entre com -1. Caso o usuário entre com um valor não-numérico, o programa deverá apresentar um erro e voltar a aguardar uma nova entrada.

10) Escreva uma função `int atoi(const char* str)` que recebe uma string contendo inteiros e retorna o valor numérico.

11) Crie uma nova função `int atoi(const string & str)` que recebe uma string contendo inteiros e retorna o valor numérico. Reaproveite a função do exercício 10.

Dica: não é necessário reescrever a função do exercício 10.

12) Escreva um programa que lê um arquivo de código em C ou C++ e remove todos os comentários, considere os comentários de bloco e de uma linha.

13) Escreva uma calculadora que utilize a notação polonesa reversa (RPN – *reverse polish notation*). Implemente as operações básicas de adição (+), subtração (-), multiplicação (x) e divisão (/). Considere as precedências das operações.

Dica: utilize a estrutura *stack* da STL.

14) Escreva um programa que converta equações em notação infixa para prefixa (RPN).

Dica: pesquise pelo algoritmo *shunting-yard*.

15) Programe uma calculadora que utilize a notação infixa.

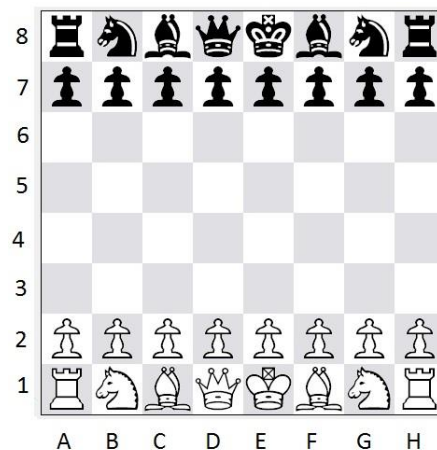
Dica: É possível reaproveitar os códigos dos exercícios 13 e 14.

16) Escreva uma calculadora lógica. Seu programa deve receber como entrada uma string de fórmula lógica, isto é, com proposições e conectivos lógicos. Implemente os conectivos de negação (~), conjunção (&), disjunção (|), implicação (>) e bicondicional (=). Considere as precedências dos conectivos. Deduza uma lista de proposições a partir da entrada e gere automaticamente a tabela verdade da fórmula.

Dica: É possível reaproveitar os códigos dos exercícios 13, 14 e 15.

17) Escreva uma função `void func(int & param)` que imprima o endereço da variável referência `param`. Defina uma variável `var` no seu main e imprima o endereço desta variável. Chame a função `func(var)` e explique o comportamento observado.

18) Considerando uma matriz **TIPO** tabuleiro[8][8]; realize o seguinte:



a) Defina uma maneira de representar, em código, os itens principais que compõem um tabuleiro: Peão, Torre, Cavalo, Bispo, Rainha, Rei e Vazio.

Dica: por enquanto, considere enumeradores e estruturas.

b) Defina uma maneira de representar a cor das peças, considere como a cor se relaciona com o jogador e com as peças.

c) Escreva uma função void iniciaTabuleiro(**TIPO** tabuleiro[8][8]).

d) Escreva uma função void entrada() que recebe uma escolha do usuário, as escolhas podem ser:

1) 'S' ou 's' para sair do jogo.

2) 'R' ou 'r' para reiniciar o jogo.

3) Uma posição de origem seguida de uma posição de destino. As posições devem seguir o padrão XY: onde X é um dígito entre '1' e '8', representando as linhas; e onde Y é uma letra entre 'a' e 'h', representando as colunas.

*) Não é necessário implementar as ações por enquanto, apenas realize a tarefa de leitura e transformação dos dados em variáveis.

e) Escreva uma função void desenhaTabuleiro(const **TIPO** tabuleiro[8][8]).

d) Utilize e adapte o seguinte modelo para seu main:

```
int main()
{
    inicializa();

    while(true)
    {
        processaEntradas();
        atualiza();
        desenha();
    }

    return 0;
}
```