

## Lista de exercícios 3

1. Faça um programa para ler uma string e após empilhar os caracteres que são letras em uma pilha P1 e empilhar os que são dígitos em outra pilha P2. Ao final, desempilhe os dados de P2 e em seguida, os de P1, imprimindo-os na tela.

### Dicas:

- Use as funções `isdigit` e `isalpha` da biblioteca `"#include <cctype>"`
  - A pilha P1 deve armazenar letras. Dessa forma, altere o arquivo `"pilha.hpp"` para atender esta necessidade.
2. Empregando pilha, crie uma função para verificar se uma expressão matemática composta apenas por parênteses está ou não *balanceada*.
    - Exemplo de expressões balanceadas: `" (A+B) "`, `" ( (A+B) + (C+D) ) "`
    - Exemplo de expressões não balanceadas: `" ( (A+B) "`, `" ) A+B ( "`, `"(A+B)) – (C + D"`

3. Desenvolva as seguintes funções:

- Função para testar se duas pilhas P1 e P2 são iguais.

**Observação:** Duas pilhas são iguais se elas possuem os mesmos elementos e na mesma ordem.

- Função para retornar o número de elementos da pilha que possuem valor par.

**Observação:** Use uma pilha auxiliar e as operações de empilha/desempilha para implementar as funções, ou seja, não é permitido acessar o vetor da pilha diretamente (`pilha->dados[i]`).

4. Faça um programa que usa uma pilha para armazenar o código das cidades que fazem parte de um caminho.
  - Inicialmente, o usuário deve informar o número de cidades;
  - Depois, o usuário deve informar o código de cada cidade e incluir na pilha;
  - Criar uma funcionalidade de busca de um código na pilha: para fazer a busca deve usar uma pilha auxiliar; a cada item verificado na pilha principal que for diferente do código procurado, deve incluir na pilha auxiliar; após encontrar o valor, deve retornar os valores da pilha auxiliar para a pilha principal.
5. Uma transportadora precisa de um sistema para gerenciar o carregamento de caixas nos caminhões da empresa. Implemente um programa que atenda a necessidade da empresa, empregando uma pilha para definir a ordem do carregamento e descarregamento das caixas em um caminhão.
  - O programa deve possuir um menu com as opções:
    - Definir quantidade de caixa – Criar uma pilha que suporte a quantidade de caixas informada.
    - Carregar caixa - empilhar
    - Descarregar caixa - desempilhar
    - Mostar carga – mostrar pilha
    - Sair do programa
  - Cada caixa possui um código e peso;  

```
struct Caixa
{
    int codigo;
    float peso;
};
```
  - Na estrutura pilha, a variável `"dados"` deve ser alterada para o tipo `Caixa`;  

```
struct Pilha
{
```

```

    int tam;
    int base;
    int topo;
    Caixa *dados;

Pilha()
{
    tam=0;
    base = -1;
    topo = -1;
    dados = NULL;
}

};

• Alterar as demais funções de Pilha.hpp para o tipo Caixa. Exemplos:
void inicializaP(Pilha *p, int tam)
{
    p->base = -1;
    p->topo = -1;
    p->tam = tam;
    p->dados = new Caixa[tam];
}

bool empilhaP(Pilha *p, Caixa c)
{
    /// retorna false se a pilha não foi inicializada ou se cheia
    if (!verificaInicializacaoP(p) || cheiaP(p))
        return false;
    else{
        p->topo++;
        p->dados[p->topo] = c;
        return true;
    }
}

```

6. A conversão de um número decimal N (base 10) para binário (base 2) é realizada por sucessivas divisões de N por 2, em que o número binário é obtido pelos restos das divisões realizadas. Crie um programa para converter número decimais em binários empregando uma pilha.

7. Crie um programa para intercalar duas pilhas P1 e P2 conforme o exemplo abaixo. Note que as pilhas P1 e P2 possuem o mesmo tamanho, mas o número de elementos armazenados em cada uma é diferente.

P1	P2	P3
2		
3	30	5
4	20	10
5	10	4
		20
		3
		30
		2