



**Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Campus Pato Branco**

Professora: Rúbia Eliza de Oliveira Schultz Ascari  
Departamento Acadêmico de Informática (Dainf)  
Tecnologia em Análise e Desenvolvimento de Sistemas  
Estruturas de Dados 2



## **Exercícios sobre Pilhas Encadeadas (Resolução)**

1. Escreva uma função que recebe 3 pilhas encadeadas, sendo que a primeira guarda n números inteiros e a segunda e a terceira estão vazias. A função deve preencher a segunda pilha com todos os números pares encontrados na primeira pilha, e deve preencher a terceira pilha com todos os números ímpares encontrados na primeira pilha.

### **itemPilhaEnc.c**

```
void separaPilhas(Pilha *p, Pilha *par, Pilha *impar) {
    Celula *aux;
    for (aux = p->topo; aux != NULL; aux = aux->prox) {
        if (aux->item.chave % 2 == 0) {
            empilha(par, aux->item.chave);
        } else {
            empilha(impar, aux->item.chave);
        }
    }
}
```

### **usaTADPilhaEnc.c**

```
printf("\nDivide pilhas: \nNumeros pares: \n");
Pilha *par = criaPilhaVazia();
Pilha *impar = criaPilhaVazia();
separaPilhas(p, par, impar);
imprimePilha(par);
printf("Numeros impares: \n");
imprimePilha(impar);

liberaPilha(par);
liberaPilha(impar);
```

2. Escreva uma função para remoção de nós duplicados (com mesma chave) em uma pilha encadeada, manipulando apenas o topo. Uma pilha auxiliar pode ser utilizada. Exemplo:

- Pilha inicial: 0, 1, 2, 2, 3, 4, 8, 9, 9
- Pilha sem duplicados: 0, 1, 2, 3, 4, 8, 9

### **itemPilhaEnc.c**

```
int buscaItemPilha(Pilha *p, int chave) {
    Celula *aux;
    for (aux = p->topo; aux != NULL; aux = aux->prox) {
        if (aux->item.chave == chave)
        {
            return chave;
        }
    }
}
```

```

    }
}
return 0;
}

void excluiDuplicados(Pilha *p)
{
    Pilha *pNova;
    pNova = criaPilhaVazia();
    //Armazena na pilha auxiliar os itens não repetidos, mas ficarão
    fora de ordem
    while (p->topo != NULL) {
        if (buscaItemPilha(pNova, p->topo->item.chave) == 0)
        {
            empilha(pNova, p->topo->item.chave);
        };
        desempilha(p);
    }
    //Insere na pilha original a partir da auxiliar para que fiquem na
    ordem correta
    Celula *aux;
    for (aux = pNova->topo; aux != NULL; aux = aux->prox)
    {
        empilha(p, aux->item.chave);
    }
    liberaPilha(pNova);
}

```

#### **usaTADPilhaEnc.c**

```

printf("\nItens da pilha antes de remover duplicados:\n");
imprimePilha(p);

excluiDuplicados(p);

printf("\nItens da pilha sem duplicados:\n");
imprimePilha(p);

```

3. Utilize uma estrutura TAD de pilha encadeada para representar um sistema para cadastrar convênios disponíveis a sócios de uma organização.
  - a) Considere que a struct a ser criada para armazenar os dados deve ser nomeada com o SEUNOME e deve ter campos referentes ao nome da empresa, área de atuação e desconto para associados.
  - b) Crie um programa que simula a criação, inclusão, exclusão e impressão dos dados sugeridos, sendo cadastrados pelo menos quatro registros de convênios.

#### Resolução:

Pode variar de acordo com os campos utilizados, mas essencialmente devem ser feito ajustes nas funções de inclusão (incluindo novos parâmetros de acordo com os campos criados) e impressão de dados (imprimindo os dados referentes aos novos campos).