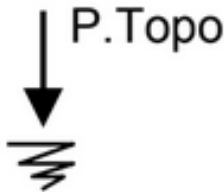
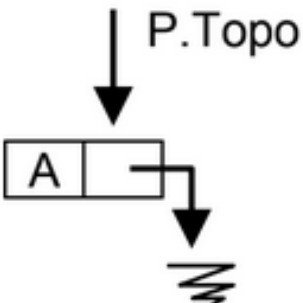
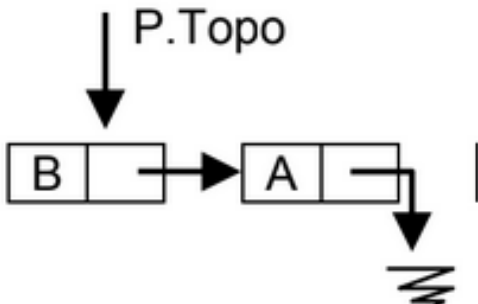
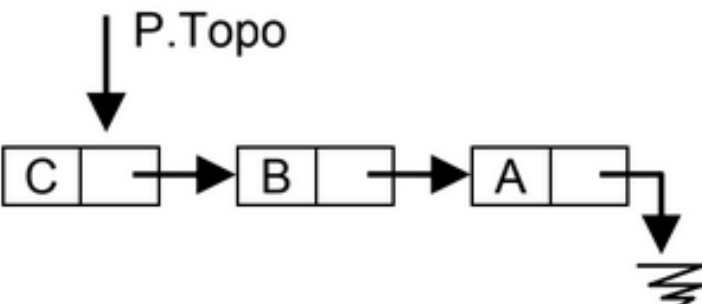
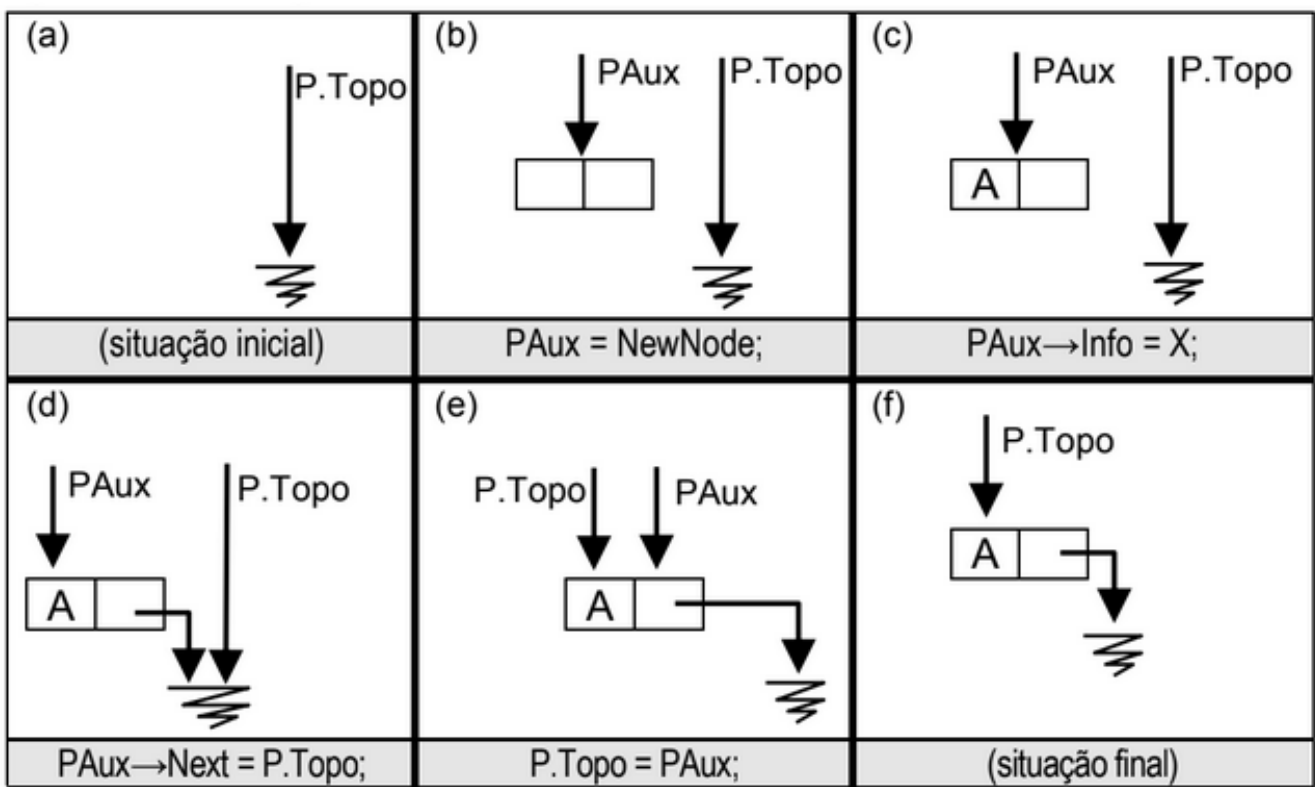
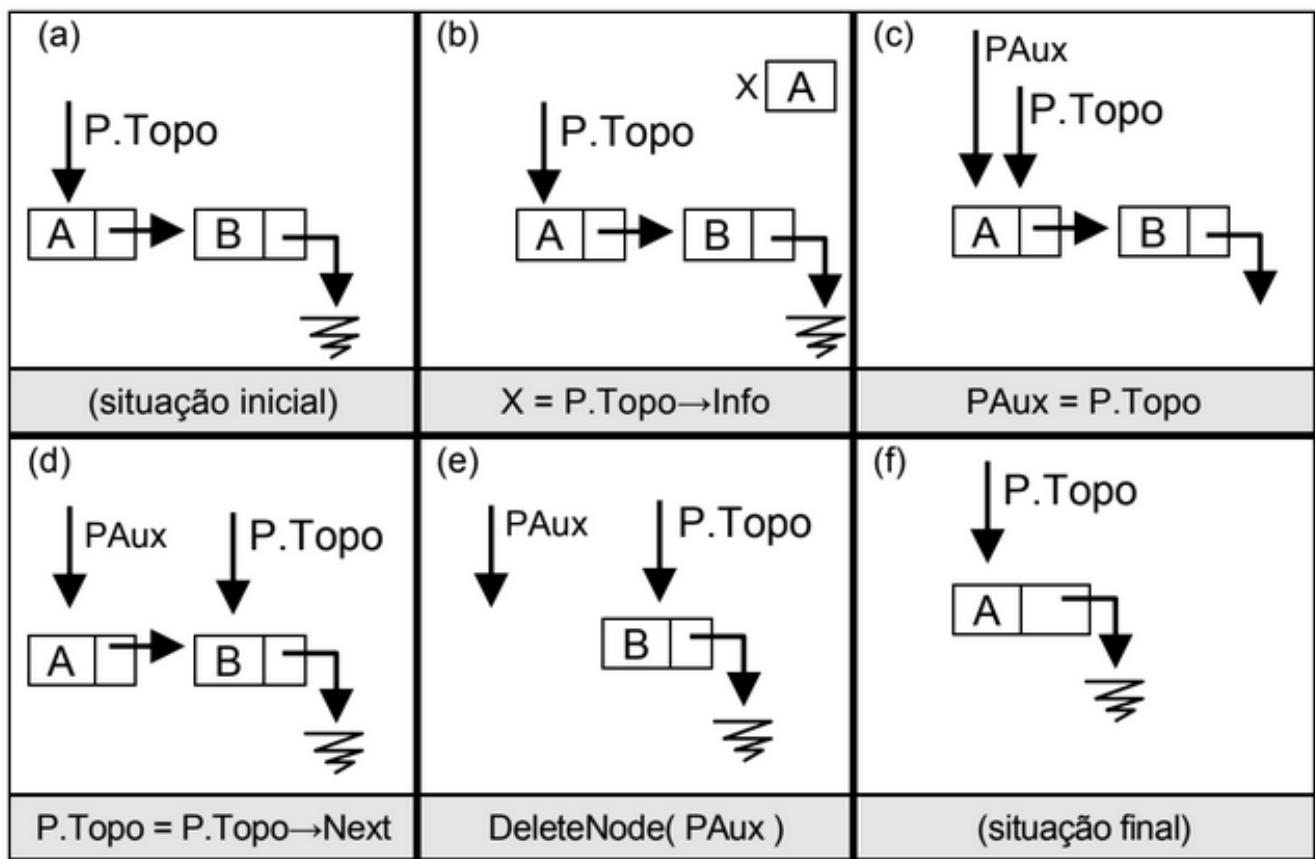


Diagramas da Pilha Encadeada

(a)	
(b)	
(c)	
(d)	
Pilha com Alocação Encadeada	



Código do arquivo pilha_encadeada.h separado em três imagens

Imagem 1

```
#include <stdlib.h>

// Declaração do Nó
typedef struct Node {
    int info;
    struct Node *next;
} Node;

// Declaração do tipo Ponteiro de Nó
typedef Node *NodePtr;

// Declaração da pilha encadeada
typedef struct Pilha {
    NodePtr topo;
    int *elementos;
} Pilha;

// Função que inicializa uma pilha vazia
void cria(Pilha *p) {
    p->topo = NULL;
}

// Função que retorna 1 se a pilha está vazia e 0 caso contrário
int vazia(Pilha *p) {
    return p->topo == NULL;
}

// Função que, nesse caso, sempre retorna que a pilha não está cheia
int cheia(Pilha *p) {
    return 0;
}
```

Imagem 2

```
// Função que adiciona um elemento à pilha caso ela não esteja cheia
void empilha(Pilha *p, int x, int *deu_certo) {
    if (cheia(p)) {
        *deu_certo = 0;
    } else {
        NodePtr p_aux = malloc(sizeof(NodePtr));

        p_aux->info = x;
        p_aux->next = p->topo;
        p->topo = p_aux;

        *deu_certo = 1;
    }
}

// Função que remove o elemento no topo da pilha caso ela não esteja vazia
void desempilha(Pilha *p, int *x, int *deu_certo) {
    if (vazia(p)) {
        *deu_certo = 0;
    } else {
        NodePtr p_aux;

        *x = p->topo->info;
        p_aux = p->topo;
        p->topo = p->topo->next;

        free(p_aux);

        *deu_certo = 1;
    }
}
```

Imagem 3

```
// Função que retorna o elemento no topo da pilha
int elemento_topo(Pilha *p, int *deu_certo) {
    if (vazia(p) == 1) {
        *deu_certo = 0;
    } else {
        int topo;

        desempilha(p, &topo, deu_certo);
        empilha(p, topo, deu_certo);

        *deu_certo = 1;

        return topo;
    }
}

// Função que retorna o número atual de elementos da pilha
int quantidade_elementos(Pilha *p) {
    Pilha p_aux;
    int x;
    int deu_certo;
    int contador = 0;

    cria(&p_aux);

    while (!vazia(p)) {
        desempilha(p, &x, &deu_certo);

        if (deu_certo) {
            contador++;
            empilha(&p_aux, x, &deu_certo);
        }
    }

    while (!vazia(&p_aux)) {
        desempilha(&p_aux, &x, &deu_certo);

        if (deu_certo) {
            empilha(p, x, &deu_certo);
        }
    }

    return contador;
}
```

Imagem 1

```
#include "pilha_encadeada.h" // Inclui a pilha encadeada

#include <stdio.h>

// Função que imprime os elementos atuais da pilha
void imprime(Pilha *p) {
    Pilha p_aux;
    int x;
    int ok;

    // Cria uma pilha auxiliar para realizar a impressões dos elementos
    cria(&p_aux);

    // Adiciona os elementos da pilha para a pilha auxiliar
    while (vazia(p) == 0) {
        desempilha(p, &x, &ok);

        if (ok) {
            empilha(&p_aux, x, &ok);
        }
    }

    printf("Imprimindo a pilha: ");

    // Imprime os elementos da pilha auxiliar enquanto até ela ficar vazia
    while (vazia(&p_aux) == 0) {
        desempilha(&p_aux, &x, &ok);

        if (ok) {
            printf("%d ", x);
            empilha(p, x, &ok);
        }
    }

    printf("← Topo\n");
}
```

Imagem 2

```
// Função main que usa as funções da pilha
int main() {
    Pilha p;
    int ok;
    int a;

    cria(&p);

    printf("\nEmpilhando 5 elementos em uma Pilha\n");

    for (int i = 10; i ≤ 50; i += 10) {
        empilha(&p, i, &ok);
        imprime(&p);
    }

    printf("Pressione uma tecla... ");
    getchar();

    printf("\nDesempilhando os elementos da Pilha\n");

    int j = 1;
    while (j ≤ 5) {
        desempilha(&p, &a, &ok);
        imprime(&p);
        j++;
    }

    printf("Pressione uma tecla... ");
    getchar();

    printf("\nAdicionando os valores 5 e 15 à Pilha\n");

    empilha(&p, 5, &ok);
    empilha(&p, 15, &ok);
    imprime(&p);

    int topo = elemento_topo(&p, &ok);
    if (ok) {
        printf("\nElemento no topo da pilha: %d\n", topo);
    }

    int total = quantidade_elementos(&p);
    printf("Total de elementos na pilha atual: %d\n", total);

    return 0;
}
```

Imagem da execução do programa

```
ranzani in AED1/Frequências/F4 on ʘ main [!?]
→ gcc usa_pilha_encadeada.c -o usa_pilha_encadeada
ranzani in AED1/Frequências/F4 on ʘ main [!?]
→ ./usa_pilha_encadeada

Empilhando 5 elementos em uma Pilha
Imprimindo a pilha: 10 <- Topo
Imprimindo a pilha: 10 20 <- Topo
Imprimindo a pilha: 10 20 30 <- Topo
Imprimindo a pilha: 10 20 30 40 <- Topo
Imprimindo a pilha: 10 20 30 40 50 <- Topo
Pressione uma tecla...

Desempilhando os elementos da Pilha
Imprimindo a pilha: 10 20 30 40 <- Topo
Imprimindo a pilha: 10 20 30 <- Topo
Imprimindo a pilha: 10 20 <- Topo
Imprimindo a pilha: 10 <- Topo
Imprimindo a pilha: <- Topo
Pressione uma tecla...

Adicionando os valores 5 e 15 à Pilha
Imprimindo a pilha: 5 15 <- Topo

Elemento no topo da pilha: 15
Total de elementos na pilha atual: 2
```