

Lista de Pilhas

1. Implemente uma função que receba uma pilha como parâmetro e retorne o valor armazenado em seu topo, restaurando o conteúdo da pilha. Essa função deve obedecer ao protótipo:

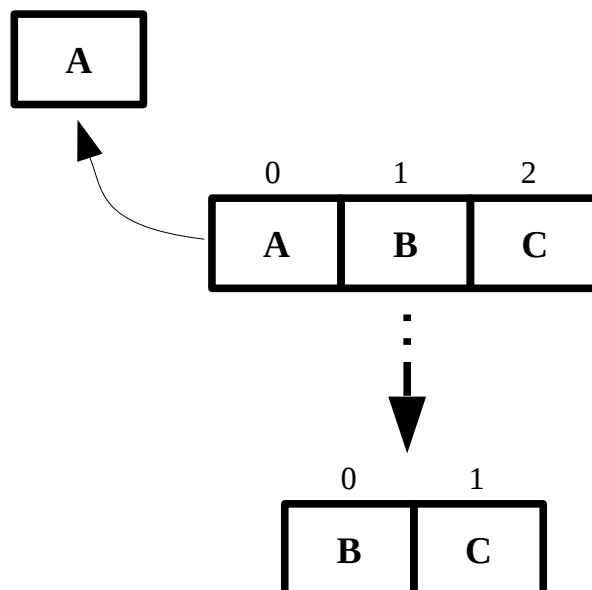
```
float topPilha(pilha *l);
```

2. Implemente uma função que receba uma pilha como parâmetro e retorne como resultado uma cópia dessa pilha. Essa função deve retornar a lista resultante, conforme ilustrado abaixo:

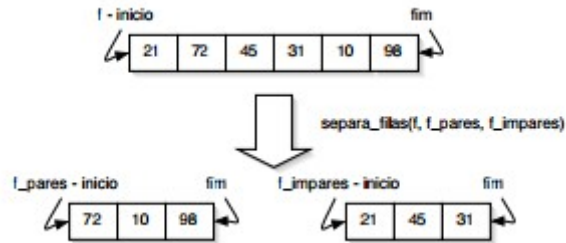
```
pilha *copiaPilha(pilha *p);
```

Ao final da função `copiaPilha`, a pilha `p` recebida como parâmetro deve ter seu conteúdo original. Essa função pode ser implementada mais facilmente com solução recursiva ou utilizando outra variável pilha auxiliar.

3. Implemente uma fila não considerado circular, onde cada operação dequeue deve deslocar para baixo todo elemento restante de uma fila. Quando essa situação ocorre e faz-se uma tentativa de inserir um elemento na fila, a fila inteira é deslocada para baixo, de modo que o primeiro elemento da fila fique na posição 0 do vetor, como mostrado na ilustração abaixo:



4. Implemente uma função que receba três filas, *f*, *fImpares* e *fPares*, e separe todos os valores guardados em *f* de tal forma que os valores pares são movidos para a fila *f_pares* e os valores ímpares para *f_impares*, conforme ilustrado a seguir:



Note que, ao final dessa função, a fila *f* vai está vazia. Essa função deve obedecer ao protótipo:

```
int separaFilas (fila* f, fila* fPares, fila* fImpares);
```